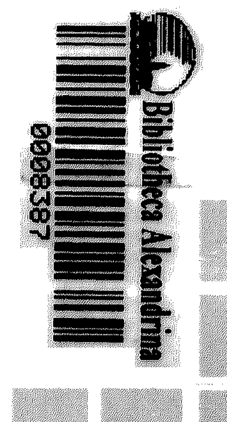
شارل برته

المساعد في المعلوماتية

ترجمة د. عبد الحسن الحسيني





المساعد في المعلوماتية

الطبعة الأولى

1408 هـ ـ 1987 م

المؤسسة الجامعية للدراسات والنشر والتوزيع

بيروت ـ الحمراء ـ شارع اميل اده ـ بناية سلام
هاتف : ٨٠٢٤٢٨ ـ ٨٠٢٤٠٧ ـ ٨٠٢٢٩٦
بيروت ـ المصيطبة ـ بناية طاهر هاتف : ٣٠١٠٣٠ ـ ٣١١٣١٠
ص . ب : ١١٣/٦٣١١ تلكس : LE ٢٠٦٦٥ ـ ٢٠٦٨٠ لبنان

سلسلة بإشراف
د. عبد الحسن الحسيني

شارل برته

المساعد في المعلوماتية

السجلات ، مجامع المعطيات ،
بنوك المعلومات ، فورتران 77 ،
كوبول ، PL / 1 ، بازيك ،
معلوماتية بعيدة .

ترجمة د. عبد الحسن الحسيني

المؤسسة الجامعية للدراسات والنشر والتوزيع

هذا الكتاب ترجمة :

AIDE - MEMOIRE D' INFORMATIQUE

Par

Charles Berthet

الفصــل الأول

أنظمة المعلومات

1 ـ نظام المعلومات

1.1 مهام نظام المعلومات

1.11 مدخل عام .

لن يفـاجأ القـارىء إذا إعتمدنـا في هذا الكتـاب التطبيقـات الخاصـة بإدارة الشـركات وتنظيمها . لأن المعلوماتية تستمد معناها من حقل إستعمالها ، ولأن هذا الاستعمال في غالبيته ، هو في حقل الإدارة والتنظيم .

المعلوماتية ليست علماً خاصاً بـالتنظيم . فهي تمتـاز كعلم باستقـلاليتها : الحـاسبات . فالأنظمة المعلوماتية لم تستطع أن تحصل على هذه القوة فوق العادية إلا بسبب جهود أولئك الذين نذروا حياتهم في خدمة الدارات الإلكترونية ، بنية الوحدات المركزية ، تقنيات الذاكرة ، مناهج أنظمة التشغيل ، المصرِّفات ، مناهج الاستعمال التي لا تُحصى ، وبشكل عام كل ما يخص هذه الأداة .

تفرض هذه الأداة ومنذ أكثر من عشرين سنة ، تأثيراً عميقاً على طريقة معالجة المعلومات ، وبالنتيجة على بنية أنظمة المعلوماتية . وهـذه تشكـل سلسلة التنظيمات التي تتحكّـم ، وتراقب دفق المواد ، المنتوجات والخدمات . هناك بعض الشركات لا تعالج عملياً سوى المعلومات : هذه هي الحـالة بـالنسبة للبنـوك ، مثلاً ، العمـلات الحجريـة والعملات الـورقية ، هي بـطبيعـة معلوماتية .

يأخذ المشاركون في أي عمل إنساني أو إستراتيجي جامع قرارتهم ، إما بالنسبة إلى مراجعهم الخاصة (إطلاقية ، ثقافية ، الخ) ، وأما بالنسبة للمعلومات التي يستلمونها . نوعية هذه الأخيرة يجب أن تكـون في أفضل حـالة . ولنفس الأسبـاب ، فإن التحكم بنـظام المعلومات لإحـدى الشركات يُصبح الوسيلة الأساسية للعناية بها في العمق . وبما ان الأداة المعلوماتية تؤثّر أكثر فأكثر على النظام المعلوماتي ككل ، فمن المهم التعرُّف عليها في جميع تفاصيلها : فهي التي تُقدِّم للشركة أو للأشخاص الخدمات الكبرى .

1.12 مهمة الانتاج

مهمة الانتاج هي أولى المهام الكبرى لنظام المعلومات . وهي تتعلّق بالتشغيل اليومي لأحد المصانع أو لإحدى التنظيمات الإدارية . نوعية المنتوجات ليست هي موضع الاهتمام : فقد تكون منتوجات مادية ، منتوجات مُصنّعة ، خدمات . . الخ . في جميع الحالات ، تدور المعلومات بشكلٍ واضح (بعكس المركز) ، من المقرِّرين نحو المُنفِّذين ، ومن الإنسان نحو الآلات . فالتنظيم يُكمل مباشرة المهمة العامة التي يتشكّل لأجلها . نضع مثلاً قطارات ، نشتري منتوجات بالجملة لتوزيعها بالمفرق ، ننقل مسافرين وبائعين ، نؤمن على الأشخاص ضد الحوادث ، ندير حركة العملات ، الخ . لذا يُنظّم نظام المعلومات الثانوي حسب قطاع العمل : أوامر إنتاج وتخطيط ، شراء مواد أولية ، إدارة المخازن ، حجز أماكن ومبيع تذاكر ، إدارة وتنظيم الملفات ، تنظيم الحاسبات الجارية . . . الخ .

هناك أنظمة ثانوية أخرى مشتركة بين جميع أشكال الأعمال ، مثلاً مدفوعات الموظفين في إحدى الشركات ، والمحاسبة التي تصحب التشغيل .

لهذا ، فالانتاج هو مفهوم « المحرِّك » في نظام المعلومات . بدون هذه المهمة ، لا يوجد أي تنظيم . ولكن الانتاج وحده لا يكفي أبداً .

1.13 مهمة التنظيم والإدارة

مراقبة حسن سير عمل إدارة أو تنظيم معين هو في غاية الأهمية ، على الأقل زمنياً أو ثانوياً . وبسبب هذه المراقبة نستطيع إكتشاف نقاط الضعف ، وإعتماد الاحتياطات للعلاج .

في هذه الحالة ، تدور المعلومات بشكل جابذ (مندفع نحو المركز Centripete) ، على عكس السابق . فهي تصعد من مختلف أعضاء التنفيذ باتجاه المُقرِّرين . ومن ثم تتركّز على جداول قيادة ، ولكنها تبقى عادة سلبية ومُطاوِعة : بسيطة ، لكن مهمة ، وعنصر قرار .

في التنظيمات ، أو في الشركات ، نحاول تمييز نوعين كبيرين من أنظمة المعلومات في الإدارة : المعلومات عن الشركة نفسها أو على حسن تشغيلها ، والمعلومات عن المحيط .

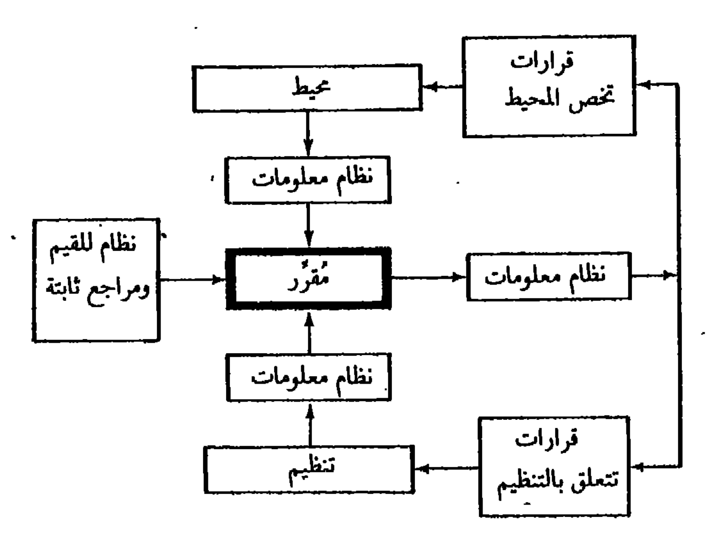
الأول هو نظير جدول القيادة ، ويحمل إستعلامات عن جميع العناصر الداخلية القادرة على إضفاء بعض الأهمية : حجم المبيعات شهر بعد شهر ، رقم العمل ، كمية المدفوعات ، حالة خزينة الشركة ، توزيع المبيعات ، الخ . المعلومات المجموعة ليست بحاجة إلى أن تكون في غاية الدقة ، ولكنها يجب أن تكون عادة ذات معنى ، جاهزة وسريعة البلوغ .

النوع الثاني هو بطبيعة أقل كمية . وهو يُعْلِم المقرِّرين عن حالة الشركة في محيطها . لهذا إذاً ، قد يكون مهماً معرفة القوة الشرائية للعائلات ، سعر الحديد ، سعر البترول ، وأيضاً قوانين الضبط ، الحالة الاجتماعية ـ السياسية المعينة ، الخ . جميع هذه المعلومات لا تدخل دائماً كأرقام ومعالجتها ليست سهلة ، فهي واسعة التأويل .

نضيف أيضاً إلى مهمة إدارة نظام المعلومات هذه ، « التوقعات » ، التي تستعمل ونظريات وتقنيات معقدة غالباً لتحديد ، مع عدم تحديد الاستعمال ، التطور المحتمل لحالة معينة . وبشكلٍ عام فتأريخ هذه الحالة يُستعمل في هذا الاتجاه .

مُخطط الكسندر (E.R. Alexander) يُوجز هاتين المهمتين الكبيرتين لنظام المعلومات في حالة التنظيمات والإدارات (شكل 1) .

وفي حالة معينة عامة في غاية السهولة ، تظهر مهمة إدارة نظام المعلومات غير مجدية : هذه هي حالة شركة شديدة التوسع ، كما هي أيضاً حالة قائد الطائرة الوحيد في السماء . في هذا التحرُّك وبدون معارضة في محيط كبير ، لا نخشى أبداً أن يُكلّفنا باهظاً ثمن الأخطاء المحتملة في الإدارة أو في القيادة .



شكل 1 ـ موقع نظام المعلومات في تنظيم معين (مخطط الكسندر المُبسَّط : وبشكل خاص ، مفاهيم ملاحظة المعلومات جرى إلغاؤها ، بالرغم من أهميتها الكبيرة ، وبالأخص بالنسبة لمعلوماتي إداري) .

ولكن عندما تضيق هذه الحالة ، فالإمكانيات تتقيَّد وتُحصر ، ومن المناسب أن نكون مستعلمين بشكل صحيح ، أي بسرعة ، بوضوح ، وبشكل كاف : هذه هي حالة قائد الطائرة بالقرب من مطار أورلي ، كما هي حالة الشركة في حالة التنافس القاسية في محيط صعب . هاتان الحالتان تتطلبان نظام معلومات فعَّال ، إن في الانتاج كما في الإدارة ، أو ، إذا فضّلنا ، في التحكم أو في المراقبة . ولنظام المعلومات مهام أخرى . فبخصوص التنظيم ، هو يحمل كثيراً من الملاحظات حول نوعية عمل المشتركين وحسوماتهم الاجتماعية . وبالتحديد ، يعني ذلك أن المعلومات الغير مفيدة يجب إهمالها حسب الإمكان ، وأنه يجب عادة إقرار القابلية في السيطرة على

المعلومات المعقدة . لن نقول هنا أكثر ، وهذه الاعتبارات تخرج عن إطار هذا الكتاب .

1.2 بنية نظام المعلومات

مهما يكن التنظيم المعتمد ، شركة ، طائرة ، كائن حي ، فهو بحاجة إلى نظام معلومات لتحريكه . هذا النظام يمكن أن ينقسم إلى ثلاثة مُركبات كبيرة : فهو يؤمن حركة الدوران ، والمعالجة وفي النهاية تخزين المعلومات .

كون المعلومات تدور فهو بديهي ، حتى إننا قد ننساه عادة . يكفي أي إضطراب في حركة الدوران هذه حتى تفقد عملية قياس المعلومات أهميتها : عطل في سنترال تلفوني ، توقف البريد ، تهديم الدارات الكهربائية في طائرة خلال طيرانها ، الخ . ضرورة نقل المعلومات من نقطة من الإدارة الى أخرى تبدأ عندما تظهر أول مرحلة من الاختلاف في هذا التنظيم ، هذا الاختلاف هو نفسه الواسطة الأمثل لتشغيل التنظيم . الذي يحاول تخصيص مجموعاته الثانوية .

تُنقل المعلومات إذاً من مكان ولادتها به إلى المكان الذي ستُعالج فيه . هذه الأمكنة يُمكن أن تكون متعدِّدة ، وهذا لا يُغيِّر أي شيء في صيغة نظام المعلومات . المعالجة تعني غالباً التقريب ، أو المقارنة مع مخططات ملائمة ، بغرض إعتماد قراراً معيناً . وهذه هي الحالة ، الأولية ، في أدوات التحكم الإلكتروميكانيكية (قيادة أوتوماتيكية مثلاً) ، كما هي أيضاً الحالة الأكثر تعقيداً في اعتماد القرار الاستراتيجي من قبل المسؤول عن الشركة .

من الواضح ، بأنه نادراً ما تستعمل المعلومات مباشرةً . يجب إذاً تحويلها ، تحضيرها ، وهذا هو قسماً من المعالجة . كما ويُمكن أن تُقارن مع معلومات أخرى ، أو تكون متكاملة أو مُركزة ، كما يمكن أن تكون ، على العكس ، متفاضلة ، الخ . وفي أفضل الأحيان ، يُمكن ألا تحتمل سوى تنسيق بسيط في الشكل في مستوى مركز المعالجة ، كي تصبح موجهة داخلياً . هذا التنسيق في الشكل هو عادة ما يكون أيضاً تغييراً في الناقل : من البيوكيمياء إلى الكهرباء ، من الكهرباء إلى السمع ، من الضوء إلى الكهرباء ، من الكهرباء إلى الرسوم ، إلخ .

وفي النهاية ، يُمكن أن تُحفظ هذه المعلومات لاستعمالات داخلية مُحدَّدة : يحتوي كل نظام معلومات على وسائط للتخزين . هناك أيضاً ، نجد جميع أشكال النواقل الفيزيائية الممكنة ، من جزئيات الحامض désoxyribo nucleique إلى الخلايا الصبغية (Chromosomes) إلى الحقول الكهرومغناطيسية الدائرة ، مروراً بالورق ، الموجود في كل مكان في التنظيمات البشرية والصعب التبديل بالرغم من كل سيئاته . أما بالنسبة لمدة الخزن فهي مُتغيِّرة من صفر وحتى ما لا نهاية ، ولا يبدو محدوداً إلا بسعات الخزن الفيزيائية ، حيث يجب إخلاء المكان اللازم من وقت إلى آخر .

فلنشير في النهاية إلى إمكانية قياس المعلومات ، ولكن فقط في بعض الحالات المحددة والدقيقة ، كما في نظرية الإتصالات البعيدة مثلاً . وفي أغلب الحالات الأخرى ، يجب أن نتكلّم عن التطور الكمي لقيمة معلومة معينة .

2 - المعالجة الآلية لنظام معلومات

2.1 - المهام

2.11 في الانتاج

إذا كان للمعلوماتية في أيامنا هذه من أهمية ، فهذا يعود لكونها الأدات الأقوى في معالجة المعلومات التي لا يستطيع الانسان إستيعابها . فهي تأخذ مداها الكامل كـأداة مثلى في أنـظمة المعلومات ، وبالتأكيد في التنظيمات . وبسبب كونها أداة للحساب صاعدة بالنسبة لكل ما كان موجوداً من قبل ، فلا يجب أن ننسى الصفة الشمولية ،وغير القابلة للتبديل في هذا النوع من التطبيقات . وليس أمامنا ، وكي نتأكد ، إلا أن نأخذ حالة السوق العالمية للحاسبات وتطبيقاتها المعلوماتية . حيث تستوعب إدارة التنظيمات من 70% إلى 80% . وحالياً بدأت المعلوماتية غير الرقمية تعرف تطوراً كبيراً .

هذا يعني ، في الانتاج ، وفي المعنى الذي حددنا به هذه المهمة ، إن المعلوماتية قد عرفت ظهورها في أنظمة المعلومات في الإدارات . وفي البداية كان ذلك يتعلّق بتطابق الميكانوغرافية الكلاسيكية مع الحاسبات . وبشكل مؤكد ، لم تعرف التطبيقات أية صفة من صفات التعقيد ، لذا أدى إدخال الحاسبات إلى خلل في توازن أنظمة المعلومات التي كنا نرغب في أتمتتها . وحتى سنة 1965 لم يكن قد بدأ السيطرة وإستيعاب الاجراءات التقنية والتنظيمية في المعالجة الآلية . وفي هذه الساعة لم تعد هذه الأمور تفرض أية مشاكل جدية .

يجب التذكير أنه في هذه المدة ، ومنذ 60-1955 ، كانالعلميون يستعملون وبشكل واسع وبدون أية صعوبة تذكر الحاسبات الأكثر قوة في زمانهم في الحساب . ولكن يجب أن نعرف أيضاً إن الاستعمال المدعو علمياً للحاسبات هو أكثر بساطة من إدخال الحاسب في نظام معلومات خاص بشركة معينة .

في الإنتاج ، نكون في حقل المعطيات المُحدّدة : ورقة الدفع يجب أن تكون دقيقة بحدود السنتمتر تقريباً ، كذلـك بالنسبـة للفاتـورة . معلوماتيـة الانتاج هي تلك الخـاصة بـالحلقات المتسلسلة ، سجلات كلاسيكية ، ولكن غالباً متسلسلة ، هي أيضاً الأعمال التطبيقيـة ، التي يُناسب كل منها مجموعة ثانوية من نظام المعلومات : إدارة المخزون ، محاسبة عامة ، إستيفاد يومي لسجلات الزبائن ، المدفوعات ، الخ . ولكن يُمكن أن تكون أكثر تطوراً ، بحيث تقوم بإدخال تقنيات بلوغ مباشر : حجز أماكن في شركة طيران ، إدارة وتنظيم الملفات في ـ شركة للتأمين .

هي أيضاً معلوماتية اللُّـغة كوبول (COBOL) ، المُنشأة لهذا النوع من الأعمال التطبيقية في سنة 1959 ، والمستعملة دائماً بالرغم من التطور الكبير الحاصل في مادة ولغات البرمجة خلال هذا القرن . معلوماتية الانتاج قادرة على نشر وإرسـال إلى عدة كيلومتـرات ، نشرات ، أو مئات الكيلوغرامات من الفواتير ، إدارة وتنظيم ، وفي غاية الدقة ، عشرات الملايين من الملفّـات في

الوقت الفعلي ، وهذا كله يتم بدون أي تدخل من جانب الانسان . ومن الممكن أن نعرف وبلحظة واحدة ، حالة أحد الحسابات المرتَّبة مع ملايين الحسابات الأخرى ، حجز مكان على طائرة باريس - نيويورك عن طريق طوكيو ، وهذا يتم في اللحظة نفسها التي نأخذ فيها قراراً بذلك ، الخ .

ولكن كل ما هو نوعي في الانتاج يصبح عكس ذلك عندما تترك المشاكل الخاصة بالإدارة مع نفس الأداة .

2.12 - في الإدارة

الإدارة هي حقل المعطيات الجاهزة بسرعة ، الواضحة وذات المعنى المُحدَّد . هي عالم الأعداد المئوية (% بالمئة) أكثر منه عالم الأعداد الدقيقة بخمسة عشر رقماً . لسنا بحاجة إلى معرفة ثمن عملية معينة بالقروش تقريباً ، ولكن بعد ثلاثة أشهر من التأخير : نتمنى تقريباً جيداً ، إذا كان ممكناً خلال النهار . ننتظر أيضاً من النظام الذي يُقدِّم المعلومات بشكل مُحدَّد وإن يكون تحضيرها وتقديمها بشكل تصبح معه قابلة للادراك والفهم من قِبل الموجهة إليه .

أنظمة المعلومات المؤتمتة الكلاسيكية ، المُتكيِّفة مع مهمة الانتاج ، تُجيب عادة بشكل سيء على حاجات الإدارة : الإحصائيات القابلة للانشاء وللتقديم بناءً على طلب المقرِّرين ، تأخذ غالباً شكل الحالات الكلاسيكية العادية للمعلوماتية : « لوائح » بحجم كبير من الأرقام ، عادةً ما تكون فائضة ، والرسوم المُحاكاة على الطابعة ، أي الصعبة القراءة ، عندما لا يتعلَّق ذلك فقط ، وفي أغلب الحالات ، بتفريغ لأحد سجلات المعطيات .

أدوات المعلوماتية المستعملة في إدارة أنظمة المعلوماتية هي أكثر تعقيداً وصعوبةً للوضع في العمل ، من تلك المُخصَّصة في الانتاج . لأنها يجب أن تجاوب على طلب شامل ، تركيبي ومفتوح : لا نعرف أبداً ، بشكل عام ، أي من المعلومات يجب أن تصعد إلى مستوى المُقرِّرين . على قدر ما تكون مهمة الانتاج محدَّدة ، تكون مهمة الإدارة بسيطة . معلوماتية الإدارة هي تلك الخاصة ببنوك المعطيات ، الوقت المُجزأ ، الأدوات الطرفية بمهام مزدوجة تصويرية وأبجعددية ، الشبكات الدولية في المعلوماتية البعيدة .

وغالباً ما نصل إلى إنشاء معلوماتية إدارة من خلال معلوماتية إنتاج جيِّدة ، وبثمن معقول نسبياً . الأصعب من ذلك هو في الاعتياد على بعض التطبيقات الجديدة ، في كل ما يتعلق برأس مال المعلومات في التنظيم .

الخبرة تؤكد أنه بعد الإنطلاق ، يصبح هذا التطور غير مرتجع : الطلب يبقى أعلى من العرض .

2.2 - البنى والتركيبات

لقد رأينا أن المركبات الكبرى الثلاث لنظام معلومات ، كانت دوران هذه

المعلومات ، ومعالجتها وخزنها .

ولقد أدت أنظمة المعلومات الأولى الداخلة في أنظمة معلومات التنظيمات الى تسريع كبير في تشغيل الأخيرتين ، المعالجة والخزن ، وأهملت الأولى . وذلك يعود لأن تكنولوجية المعلوماتية البعيدة لم تكن قد تقدمت كثيراً ، بالنسبة لتكنولوجيا الوحدات المركزية ووسائل الخزن ، من أشرطة مغناطيسية وإسطوانات . النتيجة المباشرة كانت كثافة جديرة بالذكر في دارات المعلوماتية : أي محترف في المعلوماتية لا يتذكر أكوام المطبوعات «lioting» المكدسة في المكاتب وغير المستعملة !

ولكن الأشياء تحسنت بعد ذلك ، وهذا ما كلّف باهظاً ، ولم يكن سهلاً إيضاحه . والآن ، خرجت هذه المشاكل من تاريخ المعلوماتية ، ولكن ليس من كل مكان تقريباً . تقنيات الوقت الفعلي والمعلوماتية البعيدة ساعدت كثيراً : في سدّ هذه الحاجات وإلغاء هذه الحالات .

2.2.1 ـ الدوران

تتم معالجة دوران المعلومات آلياً بسبب تقنيات شبكة المعلوماتية البعيدة : وسيكون موضع بحث واسع في هذا الكتاب . ولقد أصبح للشركات شبكاتها الخاصة ، مع خطوط مستأجرة بشكل دائم . ومن الممكن أن نحصل على بلوغ الى السجلات من بعيد وبواسطة أدوات طرفية أو معابر . هذه الأخيرة يمكن أن تكون موصولة بواسطة شبكة تلفونية عامة . ويعتمد التنظيم يومياً على مجموعة واحدة من السجلات ، تستخدم كمرجع لجميع الناس في الإدارة . وغالباً نتفادى الإزدواجية في نسخ السجلات التي تؤدي دائماً إلى واقع مبهم .

من الممكن أيضاً نقل المعطيات من النوع الرقمي على الخطوط التلغرافية ، التلفونية ، الحزم الهرتزية المباشرة أو من خلال الأقمار الاصطناعية ، الخ : أما الأجهزة المناسبة للعمل في مستوى الحاسبات ، أو أيضاً تلك الخاصة بالمستعملين ، فهي جاهزة بالكامل : موديم (مضمِّم ـ كاشف) ، أدوات طرفية ، مُركِّزات ، إضافة الى المناهج المناسبة .

ومما نلاحظه غالباً أنه منذ عدة سنوات فإن أغلب مصمِّمي الحاسبات كانوا مهتمين عن قرب بالتنظيمات الدولية في الإتصالات البعيدة . البعض الآخر كان يهتم بأهمية مفهوم الدوران في أنظمة المعلومات .

2.22 ـ المعالجة

المعالجة هي حقل الوحدات المركزية العامة أو الخاصة . وتاريخياً من هنا بدأ وجود علم المعلوماتية . فأولى الحاسبات كانت عبارة عن مكنات للحساب . أما تقنيات البلوغ إلى السجلات الأكثر تعقيداً فقد وصلت متأخرة .

وفي هذا الحقل ظهر التطور سريعاً . فلا يوجد أي شيء مشترك بين الوحدات المركزية الحالية مع تلك الوحدات التي كانت موجودة في سنة 1960 . والآن يُصنع بثمن ألف مرّة أقل ، وحدات مركزية صغيرة ، بحجم قطعة السكر ، معادلة بالكامل لتلك التي كانت موجودة منذ 15

سنة كحاسب علمي بحجم وسطي وفعّال نسبياً . ولا نزال بعيدين حتى الآن عن إمكانية إختزال حجم الوحدات المحيطية بنفس النسبة .

من المحتمل أن يكون باكراً إجراء قياس حقيقي لما حملته التكنولوجيا إلى أنظمة المعلومات . هناك شيء واضح حتى الآن هو : لم يكن ، فقط في حالة خطأ في الفهم ، لمستوى المعالجة أي تداخل أو تأثير في مستوى مجموعة المعلوماتية في أي وقت من الأوقات . فأغلب الوحدات المركزية المستعملة في إدارة التنظيمات هي بقوة فائضة وغير مستعملة بقدرتها الكاملة في عملية الحساب .

يجب أن نلاحظ أيضاً إن مناهج الاستعمال ، وبالتحديد لغات البرمجة ، قد تطورت كثيراً بالنسبة للوحدات المركزية نفسها . فنحن نبرمج كثيراً بلغة كوبول وبعد أكثر من عشرين سنة على ظهور هذه اللغة ، ويجب أن نعترف بصراحة بأن هذه اللغة هي كافية في أغلب الحالات . وباستطاعتنا أن نقوم بنفس الملاحظات في كل ما يتعلّق بلغة البرمجة العلمية FORTRAN ، التي تبلغ من العمر حتى الآن حوالي 27 سنة .

2.23 ـ الخزن

مشكلة الخزن هي معروفة من قبل جميع محترفي المعلوماتية الإدارية . وطويلاً لم يكن يُفكّر هؤلاء إلا بالأشرطة المغناطيسية ، أي بالسجلات المتسلسلة . أما سجلات الأسطوانات الحالية فلم تكن سوى تطابق لها .

ظهور الأسطوانات ، المتأخر نسبياً ، ضاعف كثيراً من إمكانيات المكنة . وباستطاعتنا الآن ، أن نصل وبسرعة الى سعة خزن تعادل مليارات السمات نبلغها مباشرة بواسطة الوحدة المركزية ، دون أن يتم تركيب أو إعادة تركيب لنواقل جديدة . مدة بلوغ المعلومات في جميع هذه النواقل هي أقل بكثير وبحدود عدة مللثواني .

وتسمح التقنيات الحالية والموجودة في الأسواق الآن ، وبمساعدة وحدات ودارات بمفاهيم جديدة ، على تخزين حتى 500 مليارد من السمات على ناقل واحد .

وبإمكان الحاسبات الكبيرة أن تُدير مجموعات كبيرة من المعطيات بسهولة : وهي متكيفة مع الحساب ومع مسائل المعلوماتية البعيدة ، ولكننا ما نزال نخشى صعوبات كثيرة ومهمة عندما نعالج سجلات كبيرة جداً . ولكن في هذه الحالات ، وكما بالنسبة لبنوك المعطيات ، فإن المجموعات المعلوماتية الكبيرة هي غير قابلة للتبديل ، مع إن الحظ يحالف حالياً الحاسبات الصغيرة .

مراجع :

كتب أساسية

Ouvrages de base :

J. ARSAC, *La science informatique*, Dunod (Paris, 1970).

R. MOREAU, *Ainsi naquit l'informatique*, Dunod (Paris, 1982).

P. TABATONI et P. JARNIOU, *Les systèmes de gestion, politiques et structures*, Presses Universitaires de France (Paris, 1975).

On pourra aussi consulter :

Ch. BERTHET et W. MERCOUROFF, *La gestion informatique*, Presses Universitaires de France, coll. Que Sais-je (Paris, nouvelle éd. 1982).

F. GALLOUEDEC-GENUYS et P. LEMOINE (sous la direction de), *Les enjeux culturels de l'informatisation* (en particulier le chapitre « *sur l'analyse des systèmes* » par M. P. SCHUTZENBERGER), Coll. informatisation et société, La Documentation Française (Paris, 1980).

J.-L. LE MOIGNE, *Les systèmes d'information dans les organisations*, Presses Universitaires de France (Paris, 1973).

On pourra enfin se reporter utilement à :

L. GALLET, *L'anglais pour informaticien*, Cedic-Nathan (Paris, nouvelle éd. 1983).



الفصــل الثــاني

تمثيل المعلومات

1 ـ أنظمة الترقيم

1.1 ـ الترقيم الثنائي

كما يعرف جميع الناس الآن ، فالتمثيل الأفضل للمعلومات في الحاسبات ، يستعمل النظام الثنائي : لا يوجد عندنا سوى إشارتين (رقمين) ، هما 0 و1 .

الكلمة (Word) هي مجموعة من الأرقام الثنائية بطول معيّـن ، يُحدِّده المُصمِّم . تنقسم الكلمة إلى عدد من السمات . إذا كان n هو طول الكلمة (أي عدد الأرقام الثنائية التي تؤلف الكلمة) ، فيمكن لهذه الكلمة أن تحتوي على عدد هو 2^n من التشكيلات المختلفة ، نفس الشيء بالنسبة للسمة : إذا كانت c هي طولها ، فيمكن للحاسب أن يستعمل مجموعة من 2^c من السمات المختلفة .

الجدول 1 يعطي مميزات بعض الحسابات الأكثر إستعمالاً . لنأخذ عدداً ثنائياً بطول n : (هنا n = 9) .

0 0 1 1 1 0 1 1 0

لنعرف قيمتها في التمثيل العشري ، نتبع الطريقة التالية . نضرب كل رقم ثنائي بـ 2^x ، x هي مرتبة الرقم الثنائي المُعتبر ، بالبدء من اليمين وبالاتّجاه نحو اليسار ، مع العدّ بدءاً من الصفر .

العدد : 0 1 1 0 1 1 1 0 0

مضروب بـ : 2^0 2^1 2^2 2^3 2^4 2^5 2^6 2^7 2^8

يعطي : $0 + 0 + 2^6 + 2^5 + 2^4 + 0 + 2^2 + 2^1 + 0 = 118.$

الجدول 1 ـ مميزات الكلمات والسمات .

أمثلة :

المكنة	طول الكلمات	طول السمات	عدد السمات كلمة
IBM 370 et 303 X	32 bits	8 bits	4
H.-B. 6000 et DPS 8	36 bits	6 et 9 bits	6 et 4
UNIVAC 1108/10	36 bits	6 bits	6
CDC CYBER 70	60 bits	6 bits	10
CII MITRA 15	16 bits	8 bits	2
DIGITAL PDP 11	16 bits	8 bits	2
DIGITAL VAX	32 bits	8 bits	4
DIGITAL PDP 10	36 bits	7 bits	5
BURROUGHS 7800	48 bits	8 bits	6 (*)
Micro-ordinateurs	8 bits	8 bits	1
	ou 16 bits	8 bits	2

الملحق C يسمح بحساب القيمة الكبرى لكلمة ، آخذين طولها بالحسبان : لكلمة بطول n ، القيمة القصوى الإيجابية الممكن أن تحتويها هي $2^n - 1$.

مثلاً ، كلمة من 15 بتة ، وعندما تكون جميع بتاتها بقيمة 1 ، فهي تمثّل الكمية 32767 (أي $2^{15} - 1$) . وكما سنرى لاحقاً ، فمن الأنسب أن نأخذ في الحسبان وجود الإشارة في كل مرّة : وفي الحقيقة لا نحصل على هذه الإشارة إلا من خلال كلمات بطول 16 بتة ، البتة الأولى هي بتة الإشارة .

1.2 ـ الأنظمة الأخرى

نظام الترقيم بقاعدة a يحتوي على a إشارة (أو عدد يعادل a من الأرقام) مُحدَّدة : هكذا فهي 2 للنظام الثنائي (0 أو 1) . a = 8 للنظام الثماني (7, 6, 5, 4, 3, 2, 1, 0) ، 10 للنـظام العشري (العشرة ارقام ، من 0 إلى 9) ، 16 للنظام السادس عشري (0 إلى F, E, D, C, B, A, 9) ، الخ .

بتعداد الأرقام من 0 إلى n − 1 ، من اليمين نحو اليسار ، وفي كلمة مؤلفة من n رقم في نظام بقاعدة a ، ونسمّي C_i الرقم ذو المستوى i (i = 0, 1..., n − 1) ، فقيمة كل عدد نحصل عليها من النتيجة التالية :

$$N = c_0 a^0 + c_1 a^1 + \cdots + c_i a^i + \cdots + c_{n-1} a^{n-1} .$$

(*) أو أيضاً 8 سمات من 6 بتات .

2 ـ المُتمِّم إلى 2 .

أغلب الحاسبات يستعمل النظام المدعو « مُتمِّم إلى 2 » ، الذي يقوم على تمثيل الأعداد الثنائية الأيجابية و« عكس » ، بطريقة ما ، السلبية منها .

يحتوي هذا النظام على عدة إيجابيات :

ـ الصفر ، لا يُمثَّل إلا بطريقة واحدة ، إيجابية .

ـ النظام هو متكيِّف مع مراصِف الدليل الجبرية .

ـ يسمح بجمع مع قسم من الكلمة ، بدون أن يكون ضرورياً أخذ بتة الإشارة بالحسبان .

ـ يسمح بالجمع والطرح مقاس 2n ، منذ أن تُعتبر الأعداد دائماً غير مؤشرة .

القواعد :

العدد الإيجابي هو عبارة عن سلسلة من الأرقام الثنائية ، حيث الرقم الأول هو دائماً 0 . فالتمثيل هو ثنائي صافي .

أمثلة :

0101	0000	الذي يعادل 5
1100	0010	الذي يعادل 44
0001	0000	الذي يعادل 1
0000	0000	الذي يعادل 0 ايجابي

العدد السلبي هو عبارة عن سلسلة من الأرقام الثنائية ، حيث الرقم الأول هو دائماً 1 . ولكي نحصل على القيمة ، يجب :

1 ـ عكس جميع البتات .

2 ـ إضافة 1 .

3 ـ تغيير الإشارة ، أي المرور من + إلى -

أمثلة :

1111	1111	الذي يعادل 1 -
0000	1000	الذي يعادل 128 -
1100	1111	الذي يعادل 4 - الخ

النظام مُتمِّم إلى 2 هو مُبهم من قِبل المبرمج . ولكن يجب أن نتذكّر وجوده عندما ندرس «Listing» (مطبعة) تفريغ الذاكرة .

3 ـ المعطيات الرقمية

3.1 ـ المعطيات الرقمية الثنائية الخالصة

3.11 مكنة IBM سلسلة 303 والمكنات بالبايتات

التمثيل يتم على 16 أو 32 بتة ، في مُتمِّم إلى 2 . البتة الأولى هي بتة الإشارة : 0 إذا كان العدد هو إيجابياً ، 1 في الحالة المعاكسة .

التوسيع :

16 بتة : من 32767 + إلى 32768 −

32 بتة : من 2147483647 + إلى 2147483648 − .

3.12 ـ هونيول بول سلسلة 6000 وDPS 8

التمثيل يتم كل 18 ، 36 أو 72 بتة ، وفي النظام « مُتمِّم إلى 2 » .

التوسيع :

18 بتة : من 131071 + إلى 131072 −

36 بتة : من $1 - 2^{35} - (\# 3,4360.10^{10})$

72 بتة : من $1 - 2^{71} +$ إلى $2^{71} - (\# 2,3612.10^{21})$

3.13 ـ يونيفاك سلسلة 1110 -1108

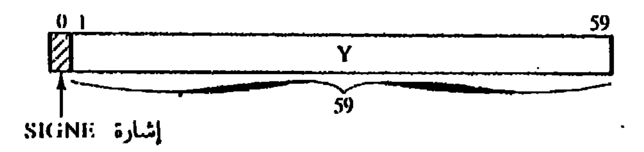
التمثيل هو شبيه بالتمثيل المُستعمل في هونيول بول والمحدَّد أعلاه .

3.14 ـ شركة Control Data سلسلة Cyber 70

يتم التمثيل في 60 بتة ، وفي مُتمِّم إلى 2 (أنظر الصورة لاحقاً) .

التوسيع :

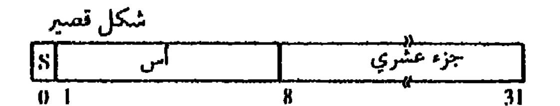
60 بتة : من $1 - 2^{59}$ إلى $2^{59} - $ (2^{59} تعادل تقريباً 6,764.1017)

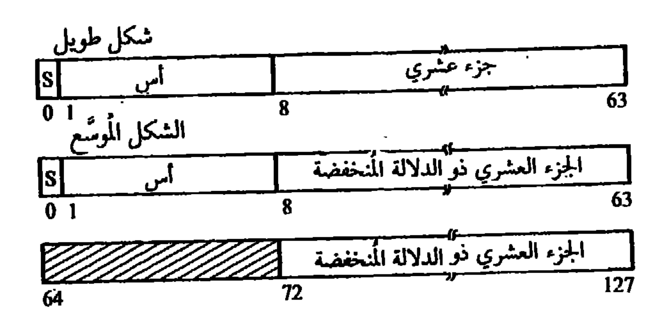


3.2 ـ التمثيل بفاصلة مُتحرِّكة

3.21 ـ IBM سلسلة 303 X والمكنات بالبايتة

يُمثَّل العدد بالشكل القصير (32 بتة) ، الطويل (64 بتة) ، أو الموسَّع (128 بتة) .





البتة الأولى تُمثِّل إشارة الجزء العشري (mantisse) : 0 للإيجابي ، 1 للسلبية .

تُمثِّل السبعة بتات التالية الأس (exposant) ، حيث يجب إضافة القيمة ، وكقيمة مؤشرة (بإشارة) ، إلى 64 كي نستطيع تمثيلها بقوة (أس) من 16 .

التوسيع :

شكل قصير : من 16^{-65} إلى $(1 - 16^{-6})\,16^{63}$

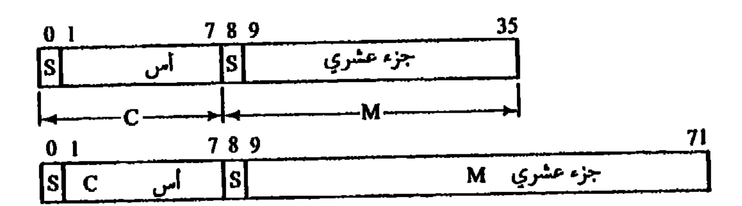
شكل طويل : من 16^{-65} إلى $(1 - 16^{-14})\,16^{63}$

شكل موسَّع : من 16^{-65} إلى $(1 - 16^{-28})\,16^{63}$

(16^{-65} تعادل $5,4.10^{-79}$ ، و 16^{63} تعادل $7,2.10^{75}$) .

3.22 ـ هونيول بول سلسلة 6000 و DPS 8

يوجد شكل قصير (36 بتة) وشكل طويل (72 بتة) .

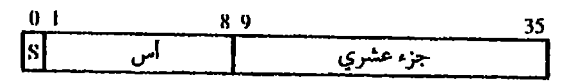


الثمانية بتات الأولى هي الأس ، في المُتمِّم إلى 2 ، والبتات الأخرى هي الجزء العشري ، وأيضاً المُتمِّم إلى 2 .

الجدول التالي يعطي التوسيعات

	الإشارة	دقة بسيطة	دقة مزدوجة
معايرة	إيجابي	$2^{-129} \leqslant N \leqslant \leqslant (1-2^{-27})\,2^{127}$	$2^{-129} \leqslant N \leqslant \leqslant (1-2^{-63})\,2^{127}$
	سلبي	$-(1+2^{-26})\,2^{-129} \geqslant \geqslant N \geqslant -2^{127}$	$-(1+2^{-62})\,2^{-129} \geqslant \geqslant N \geqslant -2^{127}$
بدون معايرة	إيجابي	$2^{-155} \leqslant N \leqslant \leqslant (1-2^{-27})\,2^{127}$	$2^{-191} \leqslant N \leqslant \leqslant (1-2^{-63})\,2^{127}$
	سلبي	$-2^{-155} \geqslant N \geqslant -2^{127}$	$-2^{-191} \geqslant N \geqslant -2^{127}$

3.23 ـ يونيفاك سلسلة 1110 -1108



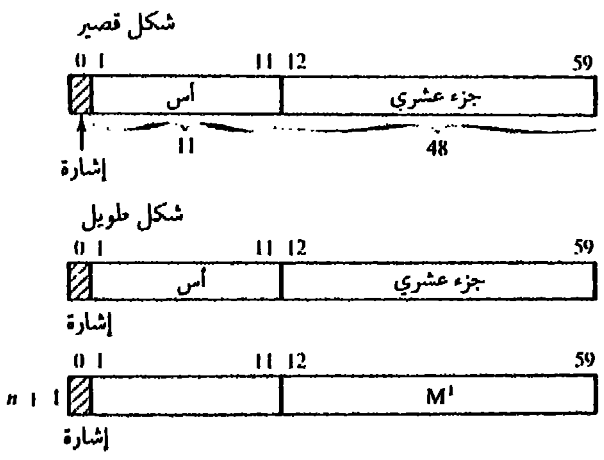
توسيع :

شكل قصير من 10^{-38} إلى 10^{-308}

شكل طويل من 10^{38} إلى 10^{308}

3.24 ـ مكنة Control Data سلسلة Cyber 70

يوجد أيضاً شكلين : قصير من 60 بتة ، وطويل من 120 بتة .



توسيع :

شكل قصير : من 10^{-293} إلى 10^{+322} (15 رقماً بدلالة)

شكل طويل : من 10^{-293} إلى 10^{+322} (29 رقماً بدلالة) .

3.3 ـ العشري الموسّع

3.31 ـ IBM سلسلة 303 X والمكنات بالبايتات

تُمثّل الرقم EBCDIC بواسطة بايتة

11000001	11000010	11000011	11000100	01000000	11110111
A	B	C	D	*b*	7

3.32 ـ هونيول بول سلسلة 6000 و DPS 8

تُمثّل الرقم BCD بواسطة 6 بتات أو ASCII بواسطة 9 بتات . هناك إذاً وحسب الحالاتِ ، 6 سمات في الكلمة أو 4 سمات في الكلمة .

3.33 ـ Controle data سلسلة Cyber 70

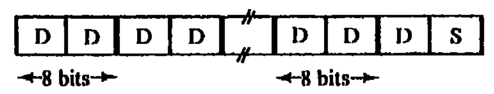
يتم التمثيل في 6 بتات . هناك إذاً 10 أرقام في الكلمة . (أنظر جدول التكويد ، الملحق B2) .

ملاحظة : في جميع الحالات ، يبقى التمثيل العشري الموسع عملي جداً في الإدارة ، وفي الحالة التي يكون فيها تفريغ الذاكرة أو السجلات مقروءاً مباشرة. ولكنه لا يقتصد في سعة الذاكرة . والطريقة الأكثر كثافةً في تمثيل المعطيات الرقمية تبقي الثنائي الصافي .

3.4 ـ العشري المكثّف

3.41 ـ IBM سلسلة 303 X والمكنات ببايتة

نقوم بتمثيل كلٍ رقمين في بايتة . وتُمثّل النصف بايتة الأخيرة الإشارة (الحرف «C» (سادس عشري) يمثّل الإشارة + ، والحرف «D» (سادس عشري) يمثّل الإشارة -) .



3.42 ـ هونيول بول سلسلة 6000

يُمثّل كل رقم في أربعة بتات في التكويد BCD . معنا إذاً 9 أرقام في الكلمة . والاتفاقات على الإشارة هي «D» (سادس عشري) للإشارة السلبية ، ومن «A» (سادس عشري) إلى «F» (سادس عشري) ما عدا «D» (سادس عشري) للإشارة الإيجابية .

4 ـ معطيات من نوع سلاسل السمات

4.1 ـ IBM سلسلة 303 X ، ومكنات بالبايتات

في التكويد EBCDIC (وفي بعض الأحيان في ASCII) ، يتم تمثيل كل سمة في

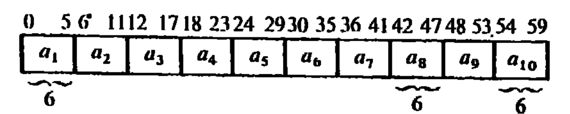
بايتة . أنظر الملحق A للائحة الأكواد .

4.2 ـ هونيول بول سلسلة 6000 و DPS 8

التمثيـل يتم بـواسـطة الكـود BCD وفي 6 بتـات أو في الكـود ASCII في 9 بتات . معنا إذاً وحسب الحالة 6 سمات في الكلمة ، أو 4 سمات في الكلمة . لِنشـر اذاً أن الإتجاه هو في تعميم التكويد ASCII (سلسلة 66) .

4.3 ـ Control Data سلسلة Cyber

يتم التمثيل في 6 بتات . معنا إذاً 10 سمات في الكلمة (أنظر جدول التكويد ، ملحق B2) .



5 ـ تدقيق في صلاحية المناهج

منذ مدة طويلة وما يزال المعلوماتيون منشغليـن بالتدقيق في صحة مناهجهم ، كما في الإدارة كذلك في الحساب الرقمي . هناك طرق عديدة موضوعة ، تهدف جميعها إلى تخفيض نسبة الأخطاء في البرامج .

إضافة لذلك ، فالبرنامج (وفي المعنى الخوارزمي) لا يقدم دائماً نتائج صحيحة : هكذا ، فعند وضعه في العمل في الحاسب ، يُنفَّـذ بواسطة دارات جبرية حيث المراصف هي بطول مُحدَّد ، وقد تقع بها أخطاء نتيجة التدوير (التكبير arrondir) . والجبر الآلي ليس سوى تقريب للجبر الدقيق .

يتعلَّـق ذلك دائماً بالحساب الجاري على الدارَّات الجبرية بفاصلة مُتحرِّكة ، والتي تكون فيها أي نتيجة مقدمة بواسطة الحاسب مربوطة بأغلاط . وقد يحدث أن هذا الخطأ هو كبير جداً بحيث تصبح النتيجة النهائية بدون معنى .

مثلاً (في الفورتران) (*) :

```
REAL X, Y, ZA, ZB
X = 1.0 E +09
Y = 1.0 E -07
ZA = (X + Y - X) / Y
ZB = (X - X + Y) / Y
```

(*) نفترض هنا إستعمال مكنة حيث الجبر بفاصلة متحركة يعطي 15 رقماً بدلالة .

من الواجب عادة إيجاد ZA = 1 و ZB = 1 . أو أن نجد ZA = 0 (غلط) و ZB = 1 (صحيح) . هكذا ، وفي حساب ZA ، العملية الأولى (X + Y) تعطي دائماً كنتيجة X ، مع بتر في الأرقام لجهة اليمين .

ولقد جرت صياغة عدة طرق لتحليل مشاكل التدوير والتكبير وإنتشارها في الحسابات . من أهم الطرق الأساسية هي طريقة ويلكنسون (Wilkinson) ، ومور (Moore) و(كيليش (Kulish) ، وبالأخص طريقة Vignes et la porte .

تدعى هذه الطريقة الأخيرة ، طريقة التبديل - التشويش ، التي أُستعملت في البداية في مسائل الجبر الخطي وبعد ذلك عُمِّمت بواسطة مخترعيها على جميع خوارزمات الحساب الرقمي . وهذه الطريقة تسمح :

- بتقييم أوتوماتيكي لانتشار وتوسع أخطاء التكبير الناتجة عن الجبر بفاصلة متحركة في المكنة ، وتقدير الدقة في النتائج الحاصلة في أي خوارزم محدَّد ودقيق .
- قطع العمليات التكرارية عندما يجري بلوغ أي حلٍّ وتقدير الدقة في هذا الحلّ .
- تقييم الخطأ في الحساب في الطرق التقريبية وتقدير الخطوة الأفضل في التقطيع .

من الممكن إذاً أن نثق بالحاسبات عند إجراء الحساب العلمي ، ولكن بشرط أن يتم التدقيق بصلاحية النتائج المقدّمة بواسطة المكنة . وهذه النتائج ستكون أكثر ملائمة طالما أن التطوير في التكنولوجيا سيعرض لنا في السوق حساسيات أكثر قوة ، تسمح لنا بكتابة وصناعة مناهج أكثر تعقيداً .

مراجع

مراجع أساسية

الوثائق التي تشرح بنية « العتاد » في الأنظمة المستعملة ، والمفروض إستشارتها في جميع الحالات .

ومن الممكن أيضاً مراجعة الكتب التالية :

G. G. BOULAYE, *La microprogrammation*, Dunod (1971).

G. BAZERQUE et C. TRULLEN, *Informatique générale*, Dunod (1971).

W. MERCOUROFF, *les ordinateurs, structure et fonctionnement des systèmes informatiques*, Cedic-Nathan (1980).

CROCUS (Nom collectif), *Systèmes d'exploitation des ordinateurs*, Dunod (1975).

Sur le contrôle de validité des logiciels numériques :

J. H. WILKINSON, *Rounding errors in algebraic processes*, Prentice Hall, 1963.

J VIGNES, M. LA PORTE, *Error analysis in computing*, Proc. IFIP Congress Stockholm 74, North Holland 1974.

J. VIGNES, *New methods for evaluating the validity of the results of mathematical computations*, Math. Comp. Simul., XX, 4, pp. 227-249, 1978.

J. VIGNES, *Une méthode pratique efficace pour évaluer la précision des résultats des algorithmes numériques*, Colloque AFCET de mars 1982 : Les mathématiques de l'informatique. (Conférence invitée.)

M. LA PORTE, J. VIGNES, *Algorithmes numériques, analyse et mise en œuvre*, Tome 1 : *arithmétique des ordinateurs et systèmes linéaires*, Technip (Paris, 1974).

J VIGNES, R. ALT, M. PICHAT, *Algorithmes numériques, analyse et mise en œuvre*, Tome 2 : *équations et systèmes non linéaires*, Technip (Paris, 1980).

الفصـــل الثـــالـــث

السجـــلات

1 ـ السجلات الفيزيائية والسجلات المنطقية

السجل هو عبارة عن مجموعة من المعطيات في المعلوماتية ، تُسجَّل هذه المجموعة بطريقة أو بأخرى على ناقل فيزيائي ، بشكلٍ تصبح معه غير مُتغيِّرة وجاهزة للاستعمال الداخلي ، إذا أوقفنا الحاسب . مفهوم السجلّ هو غير أليف إلا في المعلوماتية الإدارية ، ولكن يجب أن لا يضيع عن نظرنا بأنه لا يوجد سوى سجلات من المعطيات ، (في المعنى الطبيعي للمصطلح) ، ولكنٍ يوجد أيضاً سجلات برامج ، أو أكثر عمومية ، سجلات من المناهج . من المناسب دائماً أن نُسمِّي سجلات فيزيائية مجموعة المعطيات المُسجَّلة على ناقل (رزمة من البطاقات ، شريط مغناطيسي ، الخ . .) . هذا السجل الفيزيائي هو مستقل عن أية معالجة وله وجود مادي ساكن ودائم بين لحظة إنشائه ولحظة تهديمه . وبشكل خاص ، يمكن أن يُحفظ ، أو أن يكون موضوع عدة عمليات قراءة ، الخ . وعلى العكس ، فالسجل المنطقي هو مجموعة من المعطيات الشبيهة بالسابقة ، والمستعملة بواسطة برنامج . هذا السجل المنطقي ، لا وجود له إلا أثناء دوران البرنامج ، وبين لحظة تنفيذ أمر الفتح (OPEN) ، ولحظة أمر الإغلاق (CLOSE) المتعلقة به . هو البرنامج الذي يضيف مفهوم (متحولة) إلى مضمون مُعيَّن لتشكيل معلومات .

مثلاً :

```
NOM = 'MARTIN'.
```

يوجد عدة أنواع من النواقل المختلفة ، من البطاقة إلى الطبل المغناطيسي ، إضافة إلى عدد من الطرق للتنظيم المنطقي لهذه المجموعات من المعطيات .

2 ـ النواقل

2.1 ـ بطاقة هولورايت

عرف جميع الناس ، بطاقة الحاسب منذ مدة طويلة . من اختراع الدكتور هارمن

هولورايت (H. Holoreith) ، من جامعة كـولومبيـا(*) . وهي عبارة عن مستـطيل من الكرتون خفيف الوزن بطول 187 ملم ، وعرض 82,5 ملم ، وسماكة تعادل حوالي 100 / 7 ملم ، مُقسَّمة إلى 12 سطراً و 80 عاموداً . يستطيع كل عامود أن يحمل سمة واحدة . سنرى في الملحق قواعد التكويد والتثقيب ، لمختلف الأنظمة(**) .

2.2 ـ الأشرطة المغناطيسية

أشرطة الحاسب المغناطيسية هي عبارة عن أشرطة ميلار ، المُغطاة بمادة مغناطيسية بدورة بطاء (hysterisis) مُربَّـعة ، وبكثافة عدة ميكرون وبعرض 1/2 بوصة ، أي 12,7 ملم . وهي تُسجَّـل ، وتُقرَأ ، إما على سبعة مسارات وإما على تسعة مسارات ، والطريقة الأخيرة هي الأوسع إنتشاراً .

2.21 ـ طريقة التسجيل

هناك طريقتان للتسجيل : الطريقة NRZ (لا عـودة إلى الصفر) والـطريقة PE (طور التضمين ، أو طور الحلودة) وهناك إتجاه للتخلي عن الطريقة الأولى لحسـاب الطريقة الثانية ، المؤكدة أكثر والأسرع .

في الطريقة NRZ ، تصل الكثافة الطولية القصوى إلى 800 BPI (Bit Per Inch) . أما الكثافة الأعلى فتستعمل الطريقة PE . لن نقوم بشرحها قبل أن نقوم بوصف العملية الفيزيائية للتسجيل .

2.22 ـ كثافة الشريط المغناطيسي

تسجل المعلومات عاموديا على الشريط ، وعلى 7 أو 9 مسارات (شكل 1) .

Piste 9	0	0	1	0	0	1	0	0	1
Piste 8	1	1	1	0	1	0	1	0	0
Piste 7	0	1	1	0	0	0	0	1	1
Piste 6	0	0	1	0	0	0	0	0	1
Piste 5	0	0	0	1	0	0	0	0	1
Piste 4	0	1	1	0	0	1	1	1	0
Piste 3	0	0	0	0	0	1	1	1	0
Piste 2	1	1	1	1	1	1	1	1	1
Piste 1	1	1	1	1	1	1	1	1	0
	A	L	P	H	A	0	1	2	+

12,7 mm

شكل 1 ـ شريط مغناطيسي

(*) هـارمن هـولـورايت كـان عضـواً في اللجنـة الإداريـة لشـركـة : Computer Tabulating Recording Compagnie ، مع شارل فلينت (Flint) وتوماس جـون واتسون (Watson) . هـذه الشركـة أصبحت منذ 1924 ، شركة IBM (International Busniss machine) .

(**) لقد جرى التخلي عملياً عن البطاقات الآن . فالتقاط وجمع المعطيات يتم في الوقت الحقيقي ، بواسطة أداة طرفية . تكتب البرامج باستعمال مُنقِّح للنصوص في صيغة العمل (time-sharing) .

الكود هو هنا الكود EBCDIC . المسار رقم 9 يحتوي على سمة ثنائية للتدقيق بالتكافؤ : توضع هذه البتة ليصبح العدد الإجمالي للبتات '1' في السمة العامودية مفرداً .

نعرّف إذاً الكثافة الطولية للشريط ، بعدد السمات المُسجّلة في كل بوصة ، أو ، ما يؤدي إلى نفس الشيء ، عدد البتات الطولية في الأنش (BPI) .

الكثافات الموجودة هي التالية : (IRG تعني فجوة بين التسجيلات Inter record Gap) .

طول	طريقة التسجيل	عدد المسارات	الكثافة في BPI
19	NRZ	7	556
de 15 à 19	NRZ	7 ou 9	800
de 10 à 16	PE	9	1 600
de 10 à 16	PE	9	6 250

بإمكاننا وبسهولة إذاً حسب سعة الشريط من 2400 قدم (أي 730m) ، الذي يُمكن أن يحتوي حتى 46 مليون سمة في كثافة تعادل حوالي 1.600 BPI .

لنشر الآن إلى أن الحاسبات تستعمل أكثر فأكثر الأشرطة من نوع PE وبكثافة إلى 6250 BPI وفي 9 مسارات .

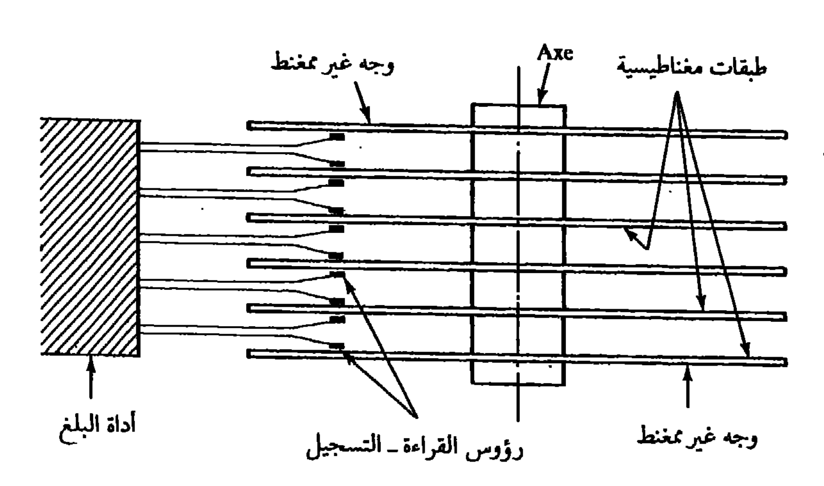
2.3 ـ الأقراص المغناطيسية (إسطوانات)

2.31 ـ وصف

تتألف الأسطوانة (القرص) المغناطيسية من ناقل ، هو عبارة عن صفيحة قاسية من الألومنيوم ، أو من سبيكة من الألومنيوم ، مغطاة وعلى وجهيها بطبقة من مادة مغناطيسية ، وبشكل عام وكما بالنسبة للشريط المغناطيسي فهناك طبقة من البرنيق (دهن صيني لامع للطلاء) تستعمل للحماية .

وبشكل عام ، هناك عدة صفائح مُكدّسة ومُثبّتة ، تؤلف ويُسمّى مكدس من الأسطوانات ، أو disk-pack . يُمكن لهذه الأسطوانات (disk-pack) أن تكون ثابتة أو متحرّكة .

يوجد أوالية بلوغ لتركيز نظام من رؤوس القراءة والتسجيل شبيهة بالرؤوس المستعملة في بسّاطة الشريط المغناطيسي . هناك دائماً رأس لكل وجه من الأسطوانة ، وقد تكون جميع الرؤوس ، وفي أي لحظة ، موجودة على نفس الخط عامودي (شكل 2) .



شكل 2 ـ رسم تخطيطي لمكدس من الأسطوانات (10 أوجه)

يرسم كل رأس قراءة ـ تسجيل مساراً دائرياً (الرأس هو ثابت خلال مدة معينة والأسطوانة تدور) . المجموعة من N مسار ، في جميع الأسطوانات (disk-pack) المؤلفة من N من الأوجه ، تناسب موقعاً معيناً لأوالية البلوغ ، وتدعى ملف إسطواني (cylindre) . من الممكن أن يكون هناك حوالي 800 أسطوانة في كل مكدس أقراص (disk-pack) معيَّن ، تناسب أكثر من 800 موقعاً مختلفاً لأداة البلوغ بالنسبة للمحور .

من المناسب أن نشير هنا إلى عدم ملامسة الرؤوس لسطح الأسطوانة : فهي تطير على علو يعادل حوالي عدة ميكرون من طبقة الطلاء الدهنية بواسطة أداة توازن نفاثة . لذا يوجد قطعة هوائية دائرة في سرعة دوران الأسطوانة . أما الرؤوس ، الشبيهة بأجنحة الطائرة ، فتطير فوق سطح الأسطوانة : وإذا حدث أي تأخير في دوران الأسطوانة نتيجة لسبب معين ، فقد يؤدي إلى أن « تهبط » الرؤوس على الأسطوانة وتؤدي إلى خرابها بالكامل . لهذا السبب يوجد أجهزة أمان تسحب الرؤوس مباشرةٍ في حالة حدوث أي حادثة من هذا النوع ، قبل أن تقوم الأسطوانة بالتأخر بشكلٍ فعلي .

إن عملية مراجعة إحدى التسجيلات من الأسطوانة تتم حسب نظام إحداثيات بأقطاب مختلفة ، يُحدَّد الشعاع بواسطة موقع الذراع وهناك أداة خاصة تقوم وبشكلٍ دائم بقياسِ الزاوية القطبية ، من 0 إلى 360 درجة ، التي تُحسب من خلال مركز معين . هذه الأداة يُمكن أن تكون مثلاً ، نظام مناضد مثبَّتاً على الأسطوانة يُراجع بواسطة خلايا كهربائية .

2.32 ـ مختلف أنواع الأسطوانات

الوحدة البسيطة في الأسطوانات هي القطاع الدائري (Sector) ، وهي عبارة عن

قطعة من المسار . عدد القطاعات في كل مسار ، إضافة إلى سعة القطاع الدائري ، يتغيّر كثيراً ، من مُصمّم إلى آخر .

2.4 ـ الطبول المغناطيسية

الطبول المغناطيسية هي عبارة عن أسطوانات معدنية مُغطّاة أيضاً بمادة مغناطيسية . رؤوس القراءة والكتابة هي ثابتة .

طريقة بلوغ التسجيلات هي شبيهة بالطريقة المستعملة في الأسطوانات : تُفرّق أيضاً هناك بين المسارات والأسطوانات .

ولقد كانت الطبول أولى النواقل المعنونة . ولكن ، هناك محاولة الآن لاستبدالها أكثر فأكثر ، بواسطة الأسطوانات (الأقراص) الثابتة ، التي تُقدّم مُميزات نظيرية ، وبالتأكيد في كل ما يتعلّق بسرعة الإرسال (أكثر من 3000 كيلو سمة في الثانية (K car/ S) . نلتقي هذه الطبول قليلاً قليلاً في التركيبات الحالية .

2.5 ـ نواقل أخرى

فلنذكر هنا الناقل C.R.A.M (Card Random Access Memory) أو ذاكرة بأوراق مغناطيسية ، في التكنولوجيا الميكانيكية الأكثر تعقيداً ، والتي تظهر في أغلب الأحيان بإمكانية عمل جيّدة (Ge MSU-388 ، IBM 2321 ، NCR CRAM 653 ، الخ) .
وهنا أيضاً ، يبدو أن الأسطوانات أكثر مضاربة .

جدول 1 ـ مراجعة موجزة عن مختلف أنواع النواقل

طريقة البلوغ	مدة البلوغ	سرعة (سمة/ثانية)	سعة (في السمات)	تكنولوجيا	الاسم
بالتوالي	99 ns	-	حتى 64 مليون سمة	M.O.S.	ذاكرة مركزية
مباشرة	10 ÷ 20ns	3 مليون	بمحور من 1 إلى 5 مليارد	مغناطيسية ثابتة	اسطوانة ثابتة
مباشرة	حوالي 20ms	1 مليون	من 200 إلى 500 مليون	مغناطيسية متحركة	اسطوانة متحركة
مباشرة	100 + 300ms	1000 + 5000	بالوحدة 1 مليون	مغناطيسية متحركة	أسطوانة لينة
بالتوالي	مُتحوّل	حتى 20000	كثافة من 250 BPI ، حتى 200 مليون	مغناطيسية متحركة	أشرطة
بالتوالي (قراءة واحدة)	متحول	من 1600	للبطاقة 80	ورق (متحرك)	بطاقات (لأوراق الذاكرة)

وغالباً ما نرى وحدات بسعة كبيرة ، تعمل كأسطوانات تُغذّى بواسطة خرطوشة مغناطيسية، تصل سعتها إلى حدود 480 مليارد من السمات (مثلاً الوحدة IBM 3850) .

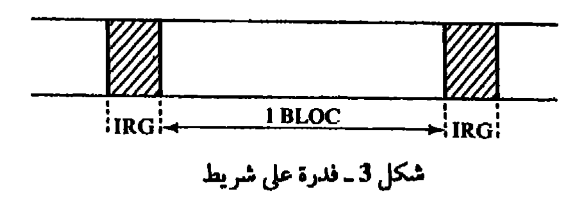
وفي النهاية ، يجب التذكير بأن العمل قد بدأ في المختبرات على ذاكرة جديدة (ذاكرة بفقاعات ، بحقل دائري ، الخ) ، ولا تبدو واعدة كثيراً بالنسبة للسرعة وحجم الخزن ، ولا تزال غير موجودة في السوق الآن .

وهذا يعني إن الإتجاه في السنوات 90-1980 سيؤدي إلى تعميم الأسطوانات الثابتة أو المتحركة .

3 ـ تنظيم السجلات

3.1 ـ الفدرات (الكتل والتسجيلات)

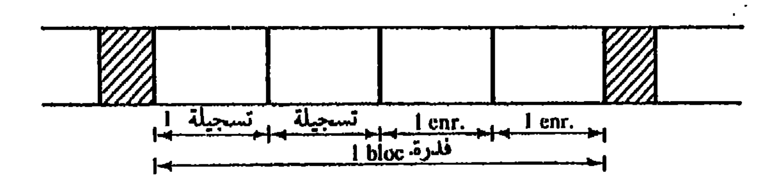
وحدة قياس المعطيات المُسجَّلة فيزيائياً هي الفدرة أو الكتلة (bloc) (شكل 3) .



شكل 3 ـ فدرة على شريط

تحتوي كل كتلة على عدد N من السمات ، هذا العدد N يُمكن أن يتغير من فدرة إلى أخرى . تنفصل الفِدرة (الكتل) عن بعضها بواسطة مساحات مُحايدة ، من 10 mm إلى 19 mm حسب التكنولوجيا المتبعة ، وتدعى TRG (Inter Records Gaps أو فجوات ما بين الفدر) . هذه الفجوات IRG ، وفي حالة إستعمال شريط مغناطيسي ، تناسب مراحل التوقف ـ وإطلاق بسًاطة الشريط .

وصف الفدرة هو عملياً البرنامج ، الذي يربط مجموعات سمات الفدرة إلى متحولات . قد يحدث إن البرنامج (كما يُفهم هنا في معنى مناهج المكنة) يُقسّم الفدرة إلى تسجيلات . (شكل 4) .



شكل 4 ـ فدرة مؤلفة من 4 تسجيلات

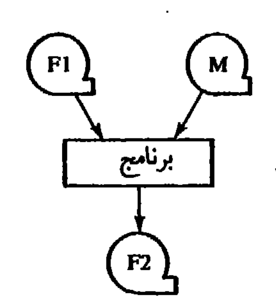
نقـول إذاً إن التسجيلات هي مُجمَّـعـة (bloqués, blocked) في n ، n هي عدد التسجيلات في الفدرة . هذا العدد n يمكن أن يكون متحولاً .

لا يوجد أي معنى لوجود التسجيلة إلا في مستوى البرنامج . فهي إذاً عبارة عن معنى مجرد في السجل المنطقي . كما ويُمكن للتسجيلة نفسها أن تكون مُقسَّـمة إلى مجموعات ثانوية ، عبارة تمثل عناصر التسجيلة .

3.2 ـ التنظيم المتتالي

التنظيم المتتالي هو الأسهل والأكثر إستعمالاً : ومن البديهي أن يُستعمل على شروط مغناطيسية . تنشيء التسجيلات الواحدة بعد الأخرى على الشريط (أو على الأسطوانة) خلال الكتابـة . إعادة القـراءة تتم أيضاً عـلى التوالي . وهـذا يؤدي ، للحصول عـلى التسجيلة رقم n ، قراءة n − 1 من التسجيلات .

الإستيفـاء اليومي يتم بـالضم (fusion, merging) لسجلَّين متتاليـين : سجـل الحركات وسجل المصدر (شكل 5) .

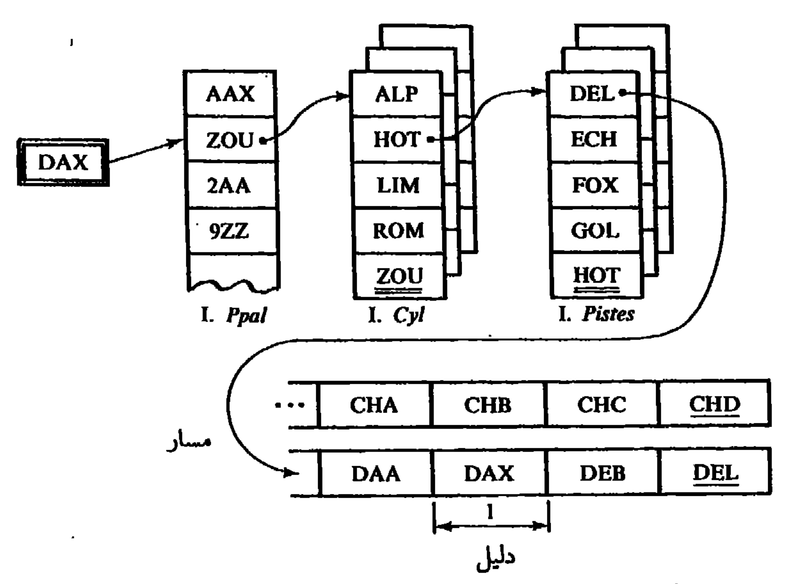


شكل 5 ـ إستيفاء يومي للسجلات المتتالية

3.3 ـ تنظيم متتالي مؤشر (بدليل) .

السجلات هي دائماً نواقل ببلوغ مباشر . نقوم بإنشاء مستويات مختلفة للدليل ، مثلاً ، في حالة الأسطوانة ، الدليل الرئيسي ، دليل الأسطوانة ، دليل المسار. تعرَّف كل تسجيلة بواسطة مفتاح ، يكون عادة قسماً من هذه التسجيلة .

عنـد إنشاء السجـل ، نكتب التسجيـلات عـلى التـوالي وحسب نـظام مفـاتيح تصاعدي . نُخزِّن في الدليل ـ المسار المفتـاح الأخير المُسجَّـل في المسـار المعتمد . بعـد ذلك ، وعندما تمتليء جميع مسارات الأسطوانة ، نُخزِّن في الدليل ـ المسار المفتاح الأخير المُسجَّـل في الأسطوانة المعتمدة . الدليل الرئيسي يُمكن أن يحتوي بدوره على مفاتيح نهائية لكل مجموعة إسطوانات (Cylindre) ، أي لكل مجموعة أقراص disk-pack (شكل 6) .



شكل 6 ـ التنظيم المتتالي بدليل

Cyl: Cylindre : إسطوانة

الإستشارة تتم حسب الطريقة التالية :

فلنفترض تسجيلة بمفتاح DAX للبحث . البحث الأول يتم في الدليل ـ الرئيسي ويدل على إن DAX ليس موجوداً في هذا الحيِّز لأن المفتاح الأخير هو AAX (لأن DAX هو « أعلى » من AAX) ، ولكنه موجود في الحيز الذي ينتهي بـ ZOU : الجدول يعطي إذاً دليل ـ الأسطوانة المعتمد . بدوره ، هذا الأخير ، وبنفس الطريقة ، يُعيد المراجعة إلى دليل ـ المسار الذي وبالنهاية يُعيد البحث إلى المسار المطلوب . يُقرأ المسار على التوالي ، حتى نصل إلى التسجيلة المطلوبة ونجدها . عند ذلك يجري إستعمالها .

الإستيفاء اليومي يتم بشكل نظيري ولكن باستعمال الحالة المستحقة لحيزات الفائض . يوجد عدد كبير من هذه الأنظمة مشتقة من صيغة البلوغ المتتالي المؤشر ، والتي تبدو مفيدة ، إذاً ضبطت بشكل صحيح مع نوع المعالجة المعتمدة .

3.4 ـ التنظيم المباشر

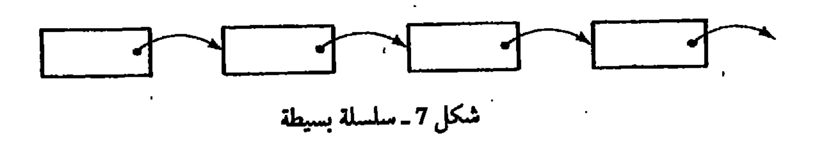
يتعلّق ذلك بالتنظيم الأساسي المباشر . تُجهَّز كل تسجيلة بواسطة مفتاح ، مُسجَّل أو غير مسجَّل . منذ البداية تقسَّم المساحة المحفوظة للسجلات إلى خلايا مُرقمة تناسب كل خلية منها تسجيلة واحدة . تتم كل عملية قراءة أو كل عملية كتابة مباشرة في الموقع المطلوب ، ويُحسب العنوان الفيزيائي للقطاع الدائري من القرص من خلال المفتاح

المُقدَّم بواسطة البرنامج . وفي نهاية المطاف ، هذا المفتاح ليس سوى الرقم التراتبي للتسجيلة في السجل (كما هو مثلاً في حالة التنظيم REGIONAL (1) في لغة PL/ 1 .

يبقى على عاتق المبرمج مهمة تنظيم السجل كما يراه ملائماً ، بخصوص عملية البلوغ المثلى (حركات أدوات البلوغ) .

3.5 ـ عمليات التنظيم الأخرى

من الممكن تصور حلول أخرى كثيرة ، منذ أن تكون المعطيات مسجلة على ناقل ببلوغ مباشر . فلنذكر مثلاً صيغة تسلسل التسجيلات . هذه الصيغة هي التالية : تحتوي كل تسجيلة على عدة مُعطيات خاصة ـ مؤشرات ـ تُمثِّل عناوين التسجيلات . مثلاً (الشكل 7) ، كل تسجيلة تحتوي على العنوان الذي يتبعه منطقياً . ولكن التوزيع الفيزيائي للتسجيلات على القرص (الأسطوانة) هو مختلف .



شكل 7 ـ سلسلة بسيطة

من الممكن وبنفس الطريقة ، إنشاء سلاسل مزدوجة ، وأيضاً متعدِّدة .

هذه الطريقة هي بسيطة ولهذا السبب نستعملها كثيراً في بنوك المعطيات. ولكن، وعلى العكس، فهي تساوي ثمناً باهظاً في المعالجة .

4 ـ حماية السجلات

الهمّ الدائم والأساسي لكل مسؤول خدمة معلوماتية ، يجب أن يكون حفظ سجلات المعطيات الحيوية . هناك عدة طرق تقنية لحلّ هذه المسألة .

4.1 ـ الحماية الفيزيائية («عتاد»)

يراقب النظام أي حادثة ويشير إليها بواسطة لوحة التحكّم أو المقراء أو الشاشة (لوحة المراقبة) .

4.11 ـ الحماية ضد الكتابة

من الممكن ، إما خارجياً (حلقات حماية الأشرطة) وإما بشكل مرتبط بالمكنة (مفاتيح إنقطاع ، أو معكاسات للوحة المراقبة) ، منع أي عملية كتابة على الناقل المعتمد . :(فلنشر إلى إن الشريط المغناطيسي هو محمي ضد أية عملية كتابة إذا كانت حلقة حمايته مرفوعة) .

4.12 ـ حماية ضد الكتابة

من الممكن إجراء عملية إعادة قراءة مباشرة لتسجيلة مكتوبة . وفي حالة عدم التوافق ، سيتم إعادة كتابة ، أو تفريغ إلى زجلة برنامج تشخيصي .

بنفس الطريقة يجري التحقق من المساحة الممغنطة قبل أي عملية تسجيل . وهناك مسارات إضافية محتملة على القرص (الأسطوانة) لاستبدال المسارات المعطّلة .

4.13 ـ مراقبة التبادل

نستعمل عادة أنظمة تحكّم بالتكافؤ (parity) . بطريقة تشبه تلك المستعملة في المعلوماتية البعيدة .

4.2 ـ الحماية المنطقية (المنهجية أو «Software»)

4.21 ـ الحماية بواسطة كلمات عبور

يوجد تدقيق بالتطابق بين هذه المعلومات المقدمة بواسطة برامج خاصة وتلك المسجّلة داخلياً في الذاكرة ..

4.22 ـ حماية في مستوى البرنامج

من الممكن تحديد طريقة استعمال السجل ، في مستوى البرنامج ، السجل لن يكون مفتوحاً إلا في القراءة ، أو في الكتابة أو في القراءة والكتابة .

في هذا المستوى (المنهاج الأساسي) ، نتوقع عادة إجراءات للمعاودة ، وبالأخص في الوقت الفعلي ، الذي يسمح بمعاودة التشغيل الجاري بدون البدء من جديد .

4.23 ـ الحماية بواسطة المدافن(Cryptage)

تحوّل سلسلة السمات التي تؤلف السجل إلى سلسلة أخرى .

برنامج المدافن هو بشكل عام كثير التعقيد . ويمكن أن يستدعي مفاتيح ، على شكل كلمات عبور ، ليست مُسجّلة أبداً ومن خلالها تتم حسابات الدفن (عدم إظهار) (Calcul de cryptage) .

لا ندفن (= رقم) سوى السجلات السرية ، التي يتوجب على كشفها تبعات كثيرة بالنسبة للشركة . ومن الأنسب إذاً ، عندما نكون في مستوى الحماية ، أن تكون خدمات التشغيل مُنظّمة بشكل دقيق في غرفة المكنة .

4.3 ـ تأمين التشغيل

تأمين التشغيل للسجلات يقوم بشكل أساسي على توقع نسخ خزن إضافية .
هذه النسخ تتم على أشرطة مغناطيسية ..
يكفي إنشاء نسخة عن السجلات الأقل أهمية أو السهلة الانشاء .

وللسجلات الكبيرة الأساسية لشركة معينة (مثلاً سجلات الزبائن ، سجلات الملفات ، سجلات المنتوجات ، الخ . .) ، فإننا نقوم عادة بالإجراءات التالية :

- نقوم بإنشاء نسخة عامة في أوقات محدَّدة وثابتة (في كل يوم ، أو في كل نهاية أسبوع) ؛

- النسخة n ، التي تدعى من الجيل الأخير ، تبقى قريبة مباشرةٍ من غرفة المكنة ، وبشكل عام في قاعة الأشرطة . تستخدم لاجراء ترميم محتمل نتيجة أي حادثة قد تحدث بالنسبة للأسطوانات .

- النسخة n - 1 يمكن أن تكون موجودة أيضاً في قاعة الأشرطة المغناطيسية . وتحفظ للحالة ، الأقل إحتمالاً ، عند تدمير النسخة n خلال عملية ترميم السجلات على القرص بواسطة هذه النسخة . من المهم أن نحفظ واحدة من النسخات أو كلتيهما في مكان مغلق ومحكم .

- وفي النهاية ، تحفظ نسخة أخرى n - 2 في مركز خاص للتأمين ، مختلف تماماً عن المركز الذي يحتوي على الخدمات المعلوماتية . وإذا كان ممكناً ، من الأفضل أن يوجد هذا المركز في الطرف الآخر من المدينة .

- وفي كل مرَّة يجري فيها إنشاء نسخة جديدة ، هناك إزاحة للأجيال . النسخة الجديدة تصبح هي النسخة رقم n ، والنسخة n تصبح النسخة n - 1 ، الخ . والنسخة n - 2 تعاد إلى مركز المكنة كي يعاود إستعمالها .

من الواضح ، أنه يعود إلى مسؤولي التشغيل ملاحظة الأخطار المرتكبة بواسطة النظام ، بالنسبة لثمن الأمان .

فنلاحظ أيضاً أنه بعد أن تتم أتمتة نظم المعلومات ، تأخذ مشاكل الأمان أهمية متزايدة ، تلمس أيضاً حقول بناء المراكز ، والتحكُّم بالبلوغ ، مراقبة الأشخاص الخ ، وإنشاء إختصاص جديد ضروري للمعلوماتيين .

مراجع

كتب أساسية

الوثائق التابعة لأنظمة السجلات الأنظمة المستعملة . من الممكن أيضاً مراجعة الكتب التالية :

C. JOUFFROY et Ch. LÉTANG. *Les fichiers. Pratique et choix de l'organisation des données informatiques*. Dunod, réédition 1981.

H. LESCA, P. BINET, *Éléments d'informatique appliquée à la gestion*, Dalloz (Paris, 1982)

Pour la cryptographie :

W. DIFFIE, M. HELLMAN, *New Directions in Cryptography*, IEEE Transactions on Information Theory, Nov. 1976.

R. L. RIVEST, A. SHAMIR, L. ADLEMAN, *A Method for obtaining Digital Signatures and Public Key Cryptosystems*, Comm. ACM, Vol. 21, N° 2, p. 120, Fév. 78.

Sur les questions politiques liées à l'utilisation des fichiers informatiques :

F. GALLOUEDEC-GENUYS et H. MAISL, *Le secret des fichiers*, Cujas (Paris, 1976).

الفصل الرابع

بنوك المعطيات

1 ـ تعريفات

1.1 ـ مجموعات السجلات

رأينا سابقاً ، في الفصل الثالث ، بماذا يتألف سجل المعلومات . فالسجلات هي ضرورية لكل نظام معلوماتي : فهي تؤلف الواسطة لأرشفة مؤقتة ودائمة للبرامج والمعطيات من كل الأنواع التي تمر عبر الحاسب . في الإدارة ، الفرّازة في المعطيات تجعل أهميتها جوهرية ، ووصف إدارة مؤتمتة كلاسيكية هو قبل أي شيء وصف للسجلات المستعملة . التنظيم الجاري هو إذاً تقسيم الأعمال الإدارية إلى أعمال تطبيقية معلوماتية ، يتعلق كل عمل تطبيقي بواحد أو عدة سجلات . هذه الأخيرة يُمكن أن تكون خاصة بتطبيق معين ، أو عادة ، مُستعملة بواسطة عدة تطبيقات مختلفة (مثلاً سجل الأشخاص في إحدى الشركات) .

أرشفة المعطيات على سجلات كُلاسيكية هو كاف في أغلب الحالات . وعادة ، في الاستعمالات الصغيرة من نوع تسلسلي ، والأشرطة المغناطيسية تلائم هذا العمل بالكامل ، وبثمن نسبي منخفض . ولكن يوجد تطبيقات مهمة ، تستعمل تقنيات في الوقت الفعلي ، التي تشتغل بشكل دقيق وصحيح مع أنظمة السجلات الكلاسيكية ببلوغ مباشر .

فوائد السجلات الكلاسيكية هي حتمية : فهي غالباً ما تكون بسيطة ، أي سهلة الاستعمال . وكل مبرمج مبتدىء يعرف وبسرعة كيفية قراءة وإنشاء سجلات متتالية . ويستطيع وبسرعة تعلّم كيفية إستعمال السجلات ببلوغ مباشر على الأقراص ، إذا كان مزوداً بلغة تعالج وبسهولة المفاتيح والتسجيلات ، مثل لغة كوبول أو PL / 1 . بعد ذلك وغالباً ، وهذا سهل التصوّر : نفكر في التسجيلات التي غالباً ما تكون ببنية ثابتة ، وعملياً بإضافة أو بعدم إضافة مفتاح بلوغ إلى كل تسجيلة ، أو ، وهذا يُعيد الأمور إلى سابق عهدها ، الاعتراف بأن السجل سيكون أو لن يكون ببلوغ عشوائي مباشر ، هو موضوع تقني بسيط . أما صيغة تنظيم المعطيات فلا تتعلّق به .

ولكن هذه الصفة الظاهرة يُمكن أن تصبح غلطة كبيرة بدون أدنى شك إذا زاد حجم المعطيات كثيراً ، كمياً ونوعياً . هكذا ، فبنية أو تركيبة السجل الكلاسيكي هي مُجمَّدة . التسجيلة تأخذ موقعاً موحداً ودائماً بين مركباتها : هي الصورة نفسها للتركيبة الوصفية للمعطيات في لغة الكوبول .

إضافة لذلك ، فالحاجة إلى تنظيم مختلف لبعض المعطيات يُجْبرنا على إنشاء سجلات جديدة من خلال عدد من السجلات الموجودة . إذا كان حجم السجلات كبيراً ، فهناك خطر من الوصول الى فائض ممنوع ، وإلى إستهلاك للنواقل مفرط ، وعلى الأخص إذا كان ذلك يتعلق بالأقراص .

إضافة لذلك ، عندما يتعلق ذلك بالسجلات المهمة لحياة التنظيم أو الإدارة (سجلات الزبائن في شركة معينة مثلاً) ، فالتركيبة أو البنية السهلة ولكن الجامدة للسجل يمكن أن تصبح طوقاً حديدياً يمنع أي تطوير داخلي . لن يكون ممكناً إضافة أي عنصر جديد غير متوقع سابقاً عند الإنشاء الأولي في أي تسجيلة ، لأن جميع البرامج ستكون مهددة بالتغيير . فقط القادرة على إحتمال هكذا تعديل ، هي الأنظمة التي تؤمِّن إستقلالية كافية بين السجلات والبرامج . وهي تستعمل ملاقي بلوغ إلى السجلات ، هذه الأخيرة لن تعالج أبداً مباشرةٍ بواسطة البرامج .

1.2 ـ مجمع المعطيات

إنّ مجموعة من السجلات الكلاسيكية يُمكن أن تحتوي على العدد الذي نرغب به مهما كان كبيراً من المعطيات المختلفة ، والعلاقات التركيبية بين هذه المعطيات هي جامدة (في سجل معين) أو غير موجودة (لسجلات مختلفة) .

إذا كانت هذه المجموعة من السجلات مُعتمدة بشكل يصبح معه من الممكن إنشاء علاقات أكثر تطوراً بين المعطيات من التركيبة البسيطة كوبول ، وإذا أمكن تعديل ، بواسطة لغة معينة ، العلاقات بين المعطيات ، فعند ذلك تدعى هذه المجموعة من السجلات مجمعاً للمعطيات ، وهذا يُؤلِّف مجموعة متماسكة من المعطيات المربوطة فيما بينها .

النظام هو إذاً مفتوح : من الممكن إضافة ، إلغاء أو إستيفاء يومي للمعطيات إضافة الى العلاقات فيما بينها .

يُخزَّن مجمع المعطيات دائماً على نواقل ببلوغ عشوائي . تعود مناهج إدارة هذه المجموعات من المعطيات ، غالباً إلى تقنيات بحث عن جداول وبواسطة حلقات تحتاج إلى البلوغ المباشر .

تعتمد عملية إنشاء مجمع للمعطيات منذ اللحظة التي يصبح فيها حجم المعطيات

المطلوب معالجتها كبيراً : وعملياً هذا يظهر بواسطة الصعوبات الملازمة لأنظمة السجلات الكلاسيكية .

وضع مجمع المعطيات في العمل هو أكثر صعوبة من إنشاء سجل إضافي في نـظام كلاسيكي : يجب إذاً تحليل العلاقات بين المعطيات . وهذا يؤدي ، عملياً ، إلى تحليل كامل للمسألة ، وهذا التحليل يمكن أن يكون صعباً .

إضافة لذلك ، يمكن لمجمع المعطيات أن يُشكِّـل القلب نفسه لإدارة التنظيم ، عند ذلك تصبح مشاكل الأمان جوهرية .

ولكن الفائدة الكبرى لمجمع المعطيات تكمن في كونه وبعد عبور المرحلة الأولى والمعقدة في فهم عملية إنشاء السجلات وترابط المعطيات، ، فهو يشكل بعد ذلك نظاماً سهلاً أغنى من مجموعات السجلات الكلاسيكية ، ويصبح أكثر سهـولة في الاستغـلال والإدارة .

1.3 ـ أنظمة إدارة مجامع المعطيات (S.G.B.D)

بينما تكفي السجلات بالتركيبة الكلاسيكية نفسها . طالما نكون مزودين بـوصف كامل للتسجيلات التي تؤلفها ، فإن مجمع المعطيات لا يُعتمد بدون المناهج الخاصة بـه للمعالجة . نفـس هذه المناهج هي التي تُميِّـز مجمع المعطيات .

هذا المنهاج يُحدِّد التركيبة (أو البنية) والمواضيع المستعملة في مجمع المعطيات . وهو يُحدِّد أيضاً وسائط الاستعمال : بشكل عام ، نجد أيضاً لغة لوصف المعطيات ولغة معالجة للمعطيات ، وإمكانيات إضافة لإمكانية القدرة على تشغيل المجمع والحماية ضد أي بلوغ غير مسموح به .

1.4 ـ بنوك المعطيات

نحتفظ غالباً بـإسم بنك المعطيات ، أو بنك المعلومات ، للمجمـوعـة المُؤلَّـفة من مجمع المعطيات ، والمناهج الخاصة بتنظيمه ، إضافة إلى البرامج الملحقة به للمعالجـة ، بواسطة الحاسب الناقل مع وسائط الاتصالات الخاصة به .

● لهذا ، نقوم بالاتصال ، غالباً بواسطة تلفون ، ببنك المعطيات : نفهم فيه تقديم وإنتاج معلومات خام أو مشغولة ، دون أن يكون هناك حاجة لأية عملية برمجة .

يمكن لبنوك المعلومات أن تكون داخلية في إحدى الشركات ، أو على العكس بمتناول الجمهور . في هذه الحالة الأخيرة ، يجري توزيعها بواسطة شبكات من الوقت المُقسَّـم .

بنوك المعلومات هي الوسيلة المعلوماتية المُميَّـزة في أنظمة المعلومات ، على قدر ما تكون داخلية في التنظيم أو مناسبة للمحيط .

2 ـ التنظيم العام لمجمع المعطيات الكلاسيكي

2.1 ـ موقع مجمع المعطيات في نظام المعلومات

يقع مجمع المعطيات على حافة نظام معلوماتي مؤتمت كلاسيكي (شكل 1) .

هكذا ، قد يحدث في أغلب الأوقات أنه يجب تكييف مجمع المعطيات هذا مع نظام موجود. هذه هي الحالة، تحديداً، حيث يكون بتصرفنا نظام فعلي جيِّد في الانتاج. فالمشكلة الصعبة في الاستيفاء اليومي للمعطيات هي محلولة جزئياً ، منذ أن تنتقل جميع المعلومات الإدارية نحو مركز الشبكة . يكفي إذاً أن نبدأ من سجلات عمل النظام ، التي هي بطبيعة متوالية (log-files) ، لإنشاء نظام ـ ثانوي أوتوماتيكي للاستيفاء اليومي لسجلات مجمع المعطيات .

وفي الحقيقة ، يجب الانتباه الى مختلف نتائج التطبيقات المتوالية حسب صيغة الانتاج الطبيعية . كما إن جميع النتائج تتحد بالاتجاه نحو مجمع المعطيات بواسطة منهاج إستيفاء يومي عام لهذا المجمع .

مهمة الإستيفاء اليومي هذه ، إضافة إلى صيانة مجمع المعطيات ، يُمكن ويجب أن تكون مستقلة عن مناهج الاستعمال أو الإستشارة بالطبيعة المختلفة تماماً ، كما سنرى تفصيلاً ولاحقاً بالنسبة لمجمع معطيات إقتصادي .

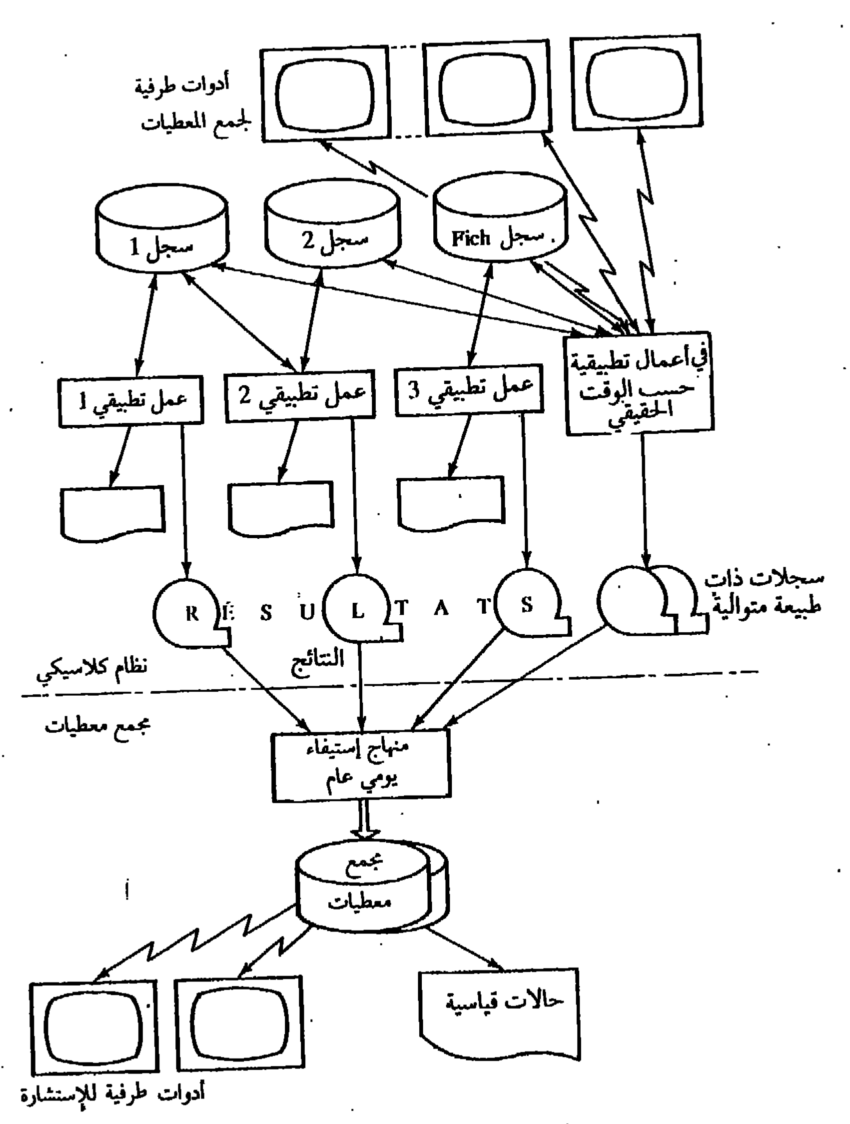
هذا التنظيم لمجمع معطيات على حافة معلوماتية كلاسيكية يمثّل حالة خاصة في أتمتة نظام معلوماتي داخلي .

تطوير هكذا حالة يتم طبيعياً نحو مجمع المعطيات بشكلٍ عام ، وفي قلب نظام المعلومات . هذا الحلّ هو نظرياً مثالي ، ولكن لا يوجد نموذج موحد ، كما أنه لا يوجد مخطط موحد لاتمتة المعلومات بشكل عام (شكل 2) .

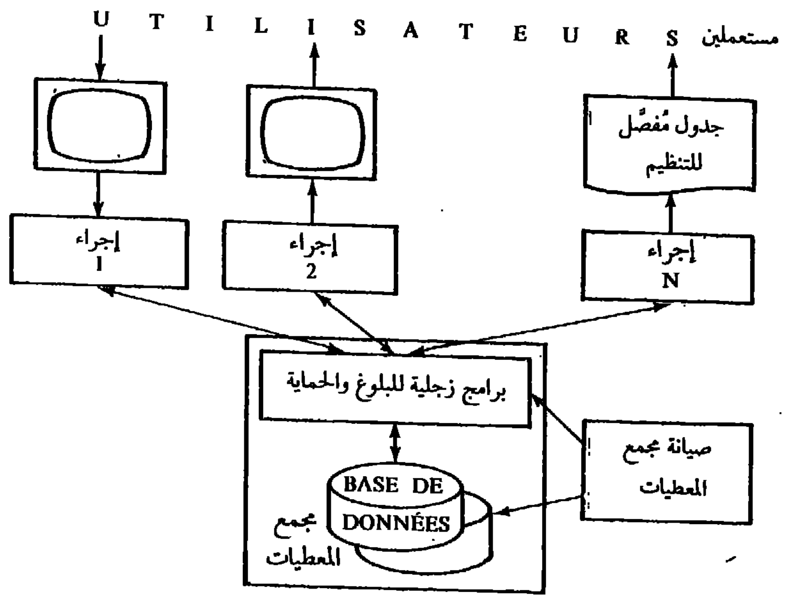
يُمكن لمجمع المعطيات أن يكون في قلب نظام معلوماتي بشرط أن لا يكون هذا الأخير معقداً . هكذا ، فإن إنشاء مجمع للمعطيات يصطدم بسرعة بالحائط التركيبي . ننتهي بأن نأخذ بالحسبان إنه إذا كان هكذا نظام ملائماً لمعيار مجموعة إدارية معينة ، فيُمكن أن يصبح غير فعال إذا أخذت هذه المجموعة بالتوسّع بشكل كبير .

2.2 ـ التركيبة العامة لمجمع المعطيات

في سنة 1959 كانت لغة كوبول تقيم إنفصالاً واضحاً بين المعطيات المُشكّلة في السجلات والبرامج ، وهذه الأخيرة كانت مفصولة بدقة ، وإلى أقسام مختلفة مناسبة للمحيط الفيزيائي للحاسبات (الأجهزة المحيطية) ، عن المعطيات نفسها ، وفي النهاية عن إجراءات المعالجة .



شكل 1 - مجمع معطيات على حافة نظام كلاسيكي



شكل 2 ـ مجمع معطيات في قلب نظام معلومات

ولقد رغبت المنظومة (*) CODASYL ، وهي في الأصل مجموعة دراسات حول الكوبول ، بتعميم هذه المفاهيم على مجامع المعطيات . والأسباب الموضوعية كانت التالية :

ـ يجب أن يسمح نظام إدارة مجمع المعطيات ببلوغ جزئي أو كلي للمجمع حسب معايير معينة .
ـ يجب أن يسمح ببلوغ متعدِّد الى المعطيات .
ـ يجب أن لا يحتوي المجمع على أي فائض .
ـ يجب أن يقوم بإنشاء إستقلال كامل بين البرامج التطبيقية ومجمع المعطيات ؛
ـ يجب أن يسمح بتركيب بنيوي لمعطيات البرنامج التي تناسب المعالجة بشكل أفضل .
ـ يجب أن تسمح بتنظيم البلوغ للمعطيات .

ولقد جرت مراجعة هذه المواضيع مراراً بواسطة مختلف المدارس التي تعالج مفاهيم مجمع المعطيات . وللمنظومة CODASYL الفضل بالحصول على تعميم شكلي ودقيق ، مدعوم بألفباء محددة (SET ، MEMBER ، الخ) ، لتعريف مجمع المعطيات .

(*) CODASYL ، Conference on Data Systems ، أنشأت سنة 1966 مجموعة عمل حول مجامع المعطيات (Data Bases Task Group) .

هناك ثلاثة مفاهيم كبرى حاضرة ، متعلقة بمجمع المعطيات :
- المواضيع المعالجة ،
- العلاقات بين المواضيع ،
- مخطط المجمع .

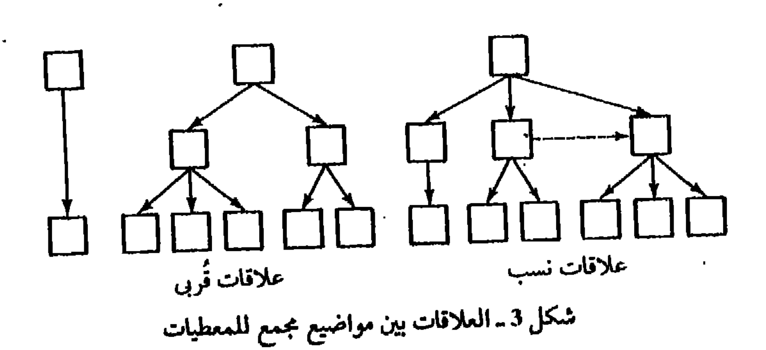
مواضيع (Objets) مجمع المعطيات هي عبارة عن المحتوى النهائي ، المحدَّد بواسطة مجموعة مركباته . مثلاً ، الموضوع « عربة » ينقسم الى « ماركة » ، « نوع » ، الخ . نفس الأمر بالنسبة إلى الموضوع « عامل » الذي ينقسم إلى « إسم » ، « إسم العائلة » ، « تاريخ الولادة » ، الخ .

نتعـرَّف هنا عـلى التطابق البسيط والصـافي بـين مفـاهيم وصف التسجيـلات في السجلات الكلاسيكية ، الناتجة عن لغة كوبول ، والشائعة عند معلوماتيي الإدارة .

بإمكاننا أن نعتبر المركبات نفسها كمواضيـع . ولكن يجب إجراء تحـديد داخـلي للعلاقات بين المركبات وهذا ما يزيد من حمل إدارة مجمع المعطيـات . إذاً هي الناحيـة العملية التي يمكن أن تقرِّر حجم التفصيل الذي يمكن أن نصل إليه ، إضافة إلى نسبة الإسهاب القصوى المقبولة .

تمثَّـل العلاقات بين المواضيع بشكل مختلف حسب المناهج . وهي عبارة عن عدة أنواع ، وذلك حسب غنى هذه المناهج .

لذا فالعلاقات المدعوة علاقات قربى هي دائماً سهلة الإنجاز ، يتعلق ذلك بعلاقة تراتبية بين موضوع وعدة مواضيع أخرى (شكل 3) . العلاقة الموجودة بـين الموضـوع ومركباته هي علاقة قربى ضمنية .



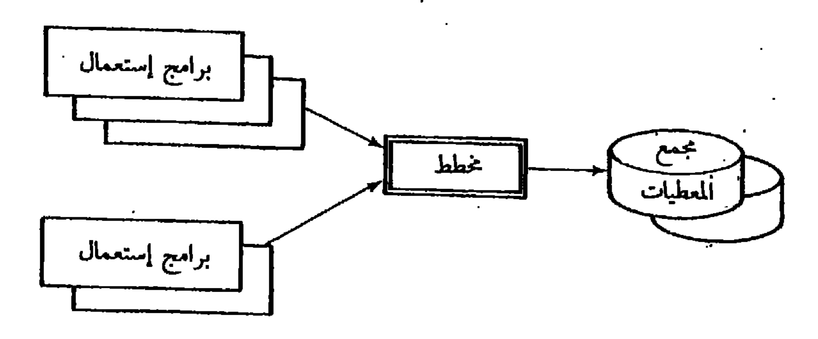
شكل 3 ـ العلاقات بين مواضيع مجمع للمعطيات

من الشائع أيضاً أن يكون بتصرفنا علاقات تدعى علاقات نسب (collatérale) . وهي تربط المواضيع بدون أن تأخذ بالحسبان التراتبية ، التي تسمح بإنشاء الشبكات .

هناك أنواع أخرى ممكنة من العلاقات : سلم إمكانيات العلاقات يؤلّف معيار إختيار بالنسبة لنظام إدارة مجمع للمعطيات .S.G.B.D .

مخطط مجمع المعطيات هو تمثيل تركيبته . هذا التمثيل يتعلّق بمواضيع المجمع ، مركباته ، العلاقات التي تربطه ، وطرق البلوغ الى هذه المواضيع ، الخ .

هذا المخطط هو نظير تقسيم المعطيات (DATA DIVISION) للغة الكوبول ، الذي يُعمّم . وهو مختلف عن برامج التطبيق ، المستقلة عن مجمع المعطيات نفسه . وعملياً ، فوجود المخطط في نظام معلوماتي حتى من نوع كلاسيكي يُقدّم إلى هذا الأخير طريقة في إستعمال مجامع المعطيات (شكل 4) .



شكل 4 ـ موقع المخطط في مجمع المعطيات

يمكن لمجامع المعطيات أن تحتوي على مواضيع خدمة ، مُولّدة أوتوماتيكياً بواسطة النظام S.G.B.D : تواريخ ، إحصائيات بلوغ إلى المواضيع ، الخ ، إضافة إلى أنظمة حماية : سماح بالبلوغ ، الخ . .

وبالنسبة لمهام مناهج إدارة مجامع المعطيات فهي تنقسم إلى مجموعتين : تلك المناسبة لانشاء المجمع ، وتلك المناسبة للتشغيل .

يُناسب كل مجموعة لغة معينة : لغة وصف المعطيات (LDD) ، ولغة معالجة المعطيات (LMD) . هذه اللغات يُمكن أن تكون متداخلة في لغة أخرى ـ أو غير متداخلة ، فهي متداخلة في لغة أخرى (فورتران ، كوبول ، PL/ 1) ، إذا كانت متكاملة نحوياً مع هذه اللغة : مثلاً ، أفعال إضافية في الكوبول ، SUBROUTINE من فورتران ، PROCEDURE في لغة PL/ 1 الخ . في الحالة المعاكسة تؤلف هذه اللغات لغة مستقلة مع نحو خاص بها .

اللغة LDD هي عبارة عن تعميم لوصف التسجيلات في السجلات الكلاسيكية .

اللغة LMD تسمح بالغاء ، إضافة ، قراءة أو تعديل معطيات المجمع ، دون لمس وصف التسجيلات .

2.3 - أمثلة على أنظمة إدارة مجمع المعطيات (*) .

يوجد عدد من أنظمة إدارة مجمع للمعطيات S.G.B.D مُوزعة حالياً ، إما بواسطة المصممين ، وإما بواسطة المستعملين أو شركات الخدمة . ولكن يجب أن نعترف بأنه قد نجد غالباً أنظمة معلوماتية كلاسيكية ومتطورة شبيهة بالنظام SGBD ، والمسؤولون عن هذا النظام إعتمدوا على مجمع للمعطيات ، كما بالنسبة للسيد JOURDAIN ، دون أن يعلموا . . .

أما مصادر النظام S.G.B.D فهي مختلفة ، البعض منها ناتج عن المنظومة CODASYL (IDS ، IDMS ، DMS-1100) ، والآخر ناتج عن تقنية اللوائح المعكوسة (MIISFIIT ، ADABAS ، SYSTEM 2000) ، والبعض الآخر أيضاً ناتج عن السجلات التراتبية (IMS ، TOTAL) . ففلسفة لغة الفول أنتجت SOCRATE .

النظام IDS هو من إنشاء .Bull-G.E ، التي أصبحت لاحقاً هونيول بول وبعد ذلك C.I.I.H.- B ، وحالياً هي فقط Bull . وهو بتصرف مناهج حاسبات Bull من السلسلة DPS 8 بنظام التشغيل GCOS . وهو يستعمل بشكل رئيسي من خلال اللغة العالية ، COBOL (التي تدعى ، لهذا السبب ، COBOL- IDS) . ولقد جرى وضعه في الخدمة في سنة 1963 ، وهو كثير الاستعمال على أغلب أنظمة ومكنات Bull .

النظام IDMS ، هو ملك Gullinane (U.S.A) . جرى توزيعه في فرنسا بواسطة الشركة S.E.M.A . ويُستعمل على مكنات شركة IBM من السلسلة 303 X ، تحت نظام التشغيل OS أو DOS ، كما ويستعمل أيضاً على مكنات SIEMENS 4004 وبعض الحاسبات الأخرى .

اللغة LDD الخاصة بالنظام IDMS هي مستقلة . وهي تقدم علاقات من نوع أقرباء وأنسباء ودائرية . أما اللغة LMD ، فهي غنية ، ومتداخلة في لغة COBOL . وضع هذا النظام في الخدمة سنة 1972 ، وهو كثير الاستعمال في العالم .

(*) إنّ الاستعلامات المقدمة في هذه الفقرة ليست سوى للإشارة والعلم . يجب على القارىء أن يراجع الكتب الخاصة بالنظام S.G.B.D في حالة الحاجة إلى التعرّف والتعمّق فيه .

DMS-1100 هو نظام S.G.B.D من UNIVAC ، الجاهز على أغلب الحاسبات الكبيـرة للسلسلة 1100 تحت نظام التشغيـل EXEC 8 . وهو متـطابق مـع تـوصيـات CODASYL .

اللغة LDD هي مستقلة ، والإستيفاء اليومي لوصف مجمع المعطيات هو ممكن ، مع إعادة تركيبه . تدخل اللغة LMD في لغة كوبول ، فورتـران وفي المؤول . ولقد وضـع DMS-1100 في الخدمة في سنة 1972 .

النـظام MIISFIIT ، جرى إنشاؤه في مـركـز الـدراسـات التقنية لتجهيـزات Aix-en-provence . ولقد اعتمد للحاسبات IBM 303 X ويعمل بالنظام VS, DOS, OS . وهو لا يُقدم علاقات أنسباء ، أو علاقات دورية . ولغته LDD هي ضمنية ، أما اللغة LMD فهي مستقلة في الصيغة التخاطبية ، أو داخلة ضمن اللغة COBOL ، أو PL/ 1 ، أو في لغة المؤول بالتوالي . ويعود تاريخه إلى سنة 1972 .

ADABAS هو صفة المناهج SOFTWARE A.G. (ألمانيا) . ويوزع في فرنسا بواسطة شركة للخدمات . العلاقات الأنسبائية والدورية هي ممكنة . لا يوجد أية معادلة للمخطط ، ولكن هناك إستقلال بين مجمع المعطيات والبرامج . يمكن للنظام ADABAS أن يعمل على الحاسبات IBM 303 X تحت إشراف نظام التشغيل OS أو DOS ، VS ، وعلى الحاسبات SIEMENS 4004 أيضاً . اللغة LDD هي مستقلة ، أما اللغة LMD فهي إما مستقلة (ADASCRIPT)، إما ضمن اللغة COBOL، FORTRAN، PL/1 أو المؤول ، أو على شكل طلبيات (CALLS) . يعود تاريخ هذا النظام إلى سنة 1974 ويبدو أنه كثير الاستعمال .

SYSTEM-2000 ، ويدعى أيضاً S2K، وهو من إنتاج MRI systems (U.S.A) corporation . ويوزع بواسطة شركات الخدمة وشركات الخدمة المكتبية . ويعمـل على أغلب الوسائط والحاسبات الكبيرة : IBM 303 X بإشراف النظام UNIVAC 1100, OS (1110, 1108, 1106) بإشراف النـظام EXEC 8 ، وControl Data من السلسلة 6000 بإشراف النظام SCOPE أو KRONOS (وحالياً : NOS) . اللغة LDD هي مستقلة (DEFINE) . أما لغة LMD فهي مستقلة أو داخلة في لغة COBOL ، FORTRAN أو المؤول . موضوع في الخدمة في سنة 1972 ، ويُستعمل كثيراً في العالم .

IMS هو من إنتاج IBM . ويُمكن أن يعمل على الحـاسبات من السلسلة 303 X بإشراف النظام OS أو DOS . وهو مجهَّـز بعلاقات قربى وعلاقات أنسباء . تتألف اللغة LDD من ماكرو ـ تعليمات (تعليمات كبْرى) . أما اللغة LMD فهي مستقلة (GIS, IQF) أو داخلة ضمن لغة COBOL ، PL/ 1 أو المؤول ، بشكل طلبيات (CALL) إلى

DL/ 1 . ويُمكن لـ DL / 1 أن يُستعمل مع CICS .

TOTAL هو من إنتاج CINCOM systems . ويبدو أنه كثير الاستعمال ، بمراعاة وفي نفس الوقت ، لعدد المكنات من جميع الأنواع الجاهز عليها IBM 303 ، SIEMENS ، X ، NCR ، Control Data ، الخ) وحاجاته القليلة من الذاكرة . اللغة LDD هي مستقلة (DBDL) ، واللغة LMD فهي داخلة بشكل طلبات (CALL) في كوبول COBOL ، PL/ 1 ، FORTRAN ، FORTRAN أو Assembler .

SOCRATE هو من إنشاء جامعة غرينوبل (IMAGE) . ولقد أقيم بعدّة إجراءات تنفيذية لهذا المنهاج من قبل .C.I.I ، وECA Automation . وهو يعمل على المكنات BULL (DPS7 ، DPS8 ، IRIS) إضافة للمكنات IBM 303 X ، بإشراف النظام DOS أو OS . وهو مُجهّز بعدد وافر من التركيبات (العلاقات) والامكانيات . اللغة LDD هي مستقلة . أما اللغة LMD فهي مستقلة وتخاطبية ، ومتداخلة في الكوبول أو المؤول المتوالي . يعود تاريخه إلى سنة 1973 ويعمل في عدد التطبيقات الإدارية المهمة .

فلنذكر أيضاً بعض الأنظمة الأكثر شهرة ، MARK IV ، QUAERY-UPDATE ، MARS VI ، FORTE ، GIM ، الخ .

يجب أن نشير أيضاً إلى أن هذه المناهج تحتاج إلى مكان كبير في الذاكرة المركزية ، كي تصبح فعالة . هكذا فالنظام SYSTEM- 2000 يحتاج إلى 164 K كحد أدنى ، أو SOCRATE فيحتاج إلى 128 KB على المكنة IRIS 50 ، والنظام IMS فيحتاج إلى أكثر من 400kb بإشراف النظام OS/ 370 ، والنظام IDS يحتاج إلى أكثر من 64 k Word (64 كيلو كلمة) من 36 بتة للكلمة الواحدة على المكنات سلسلة DDS 8 ، الخ .

3 ـ بنوك المعطيات الاقتصادية

بنوك المعطيات الاقتصادية تجمع ، على الحاسبات ، معلومات مهمة كما بالنسبة للتشغيل الداخلي للشركات فكذلك بالنسبة للمحيط . فهي إذاً حالة خاصة مهمة من بنوك المعطيات بشكل عام وتؤلف الأداة المميّزة لمهمة إدارة أنظمة المعلومات .

وهي تُستخدم لانتاج ولتقديم ، وفي شروط السرعة والدقة وبأشكال كافية ، المعلومات الخام أو المشغولة الضرورية للقرارات الاستراتيجية أو لإدارة مُتعلّقة بشركة معينة .

فهي صورة للحالة الحاضرة لشركة معينة ومحيطها الاقتصادي، إضافة إلى تأريخها. ويُمكن أن تتدخّل في عمليات التوقعات ، لأمد طويل أو قصير .

تتمتع بنوك المعطيات الداخلية في الشركات بنفس الطبيعة، من وجهة نظر

معلوماتية ، كبنوك المعطيات في المحيط ، والتي تدعى بنوك معطيات إقتصادية ـ كبرية (macro-economique) .

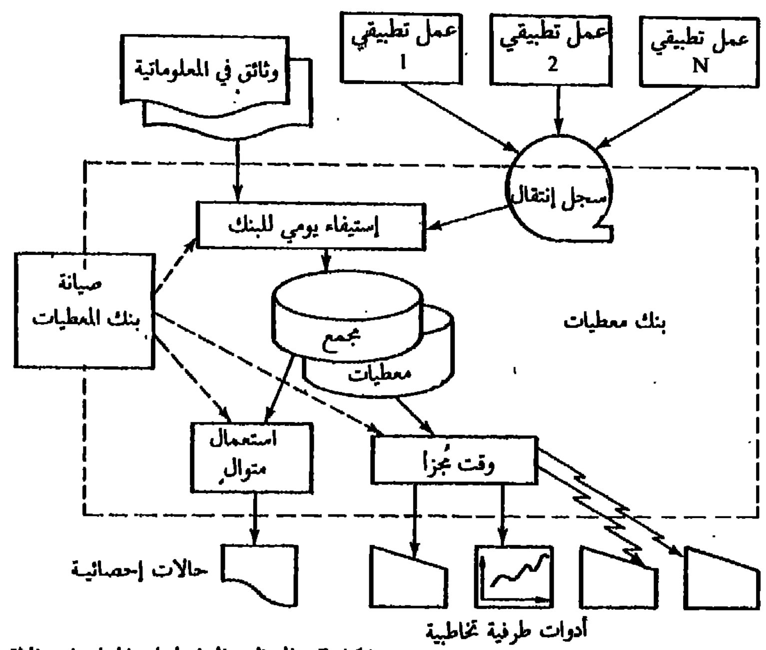
3.1 ـ تنظيم بنك معطيات إقتصادية(*)

3.11 ـ عموميات

يقع بنك المعطيات الاقتصادية داخل إحدى الشركات على حافة نظام المعلومات . وهو يُشكل نظاماً ثانوياً مُطاوعاً وغير فعال بالنسبة لمهمة الانتاج ، التي يجمع المعلومات فيها .

يتألف بنك المعطيات من مجمع معطيات مختلف متحولات إدارة الشركة : حالة المخزون في السلع ، عدد وثمن حالة المبيعات بالمنتوجة وبالقطاع ، الخزينة ، الطلبيات ، الخ . . .

عدد المتحولات المختلفة يُمكن أن يكون مرتفعاً ، وإختيار المتحولات التي يجب أن تدخل إلى بنك المعطيات هي مشكلة صعبة للإدراك . هكذا ، فهو نظام مفتوح ، قابل للتعديلات : من الممكن أن لا نتوقع خزن إحدى المتحولات بينما معرفتها هي ضرورية في لحظة معينة



شكل 5 ـ التنظيم العام لبنك المعلومات الاقتصادي

(*) المعلومات الواردة في هذه الفقرة هي ذات طبيعة إعلامية : وناتجة عن خبرة المؤلف في هذا المجال . ولكن يوجد طرق أخرى فعالة ، عملياً على بنوك المعطيات الاقتصادية .

تُسجَّـل مختلف قيم المتحولات دورياً ، لتُشكِّل سلسلية تأريخية . معـرفة القيمـة الحالية لمتحولة معينة وتاريخها في مدة طويلة تسمح ، في بعض الحالات ، بالتعميم على مدة زمنية قصيرة وإجراء توقعات . يجب أن لا ننسى الصفة السريعة العشوائية لهكذا عملية ، إذاً لم تُحاط بأكثر الاحتمالات . وعملياً ، فالتوقعات الاقتصادية هي شديدة التعقيـد ، وتستدعي بشكل خـاص ، تصاميم نـظرية بـطبيعة إحصـائية وإقتصـادية . وبـإمكانها بالتحديد ، الإرتكاز على بنك معطيات إقتصادي كناقل للنماذج المستعملة .

مهمة مجموعة البرامج والاجراءات هي الاستيفاء اليومي لمجمع المعطيات . يتعلَّق ذلك بالقسم الأهم في النظام : بدون إستيفاء يومي صحيح ومنتظم ، يفقد المجمع قيمته بسرعة .

الإستيفاء اليومي للمجمع يمكن أن يكون عملية مؤتمتة بالكامل ؛ أو يدوية ، أو يدوية ـ أوتوماتيكية . فقط تركيبة نظام المعلومات تُحدِّد الطريقة التي يتم فيها هذا الاستيفاء اليـومـي . والحلُّ الأفضل يقوم على إستيفاء يومي مشترك يدوي ـ أوتوماتيكي ، مع نسبة كبيرة منها مؤتمتة .

إستعمال مجمع المعطيات يؤدي إلى تدخل تقنيات الوقت المجزأ (المقسَّـم) (time sharing) . وفي المقياس الذي تكون فيه هذه التقنيات والوسائط ممكنة وجاهزة ، فهي تزودنا بإمكانيات بلوغ سهلة وفعالة . وعملياً ، لا يجب أن نتصوَّر إمكانية وضع مجمع للعمل داخل إحدى الشركات بدون أن يكون بإمكاننا أن نحصل على نظام تشغيـل في الوقت المقسَّـم لاستعماله .

من الضروري ، إضافة لذلك ، أن يكون استعمال بنك المعطيات مختلفاً بالكامل عن إستيفائه اليومي .

هكذا ، فالاستيفاء اليومي هو عملية مركز خدمة موحدة ، ومُنظَّم بالكامل لهـذه الغاية : الخدمة المعلوماتية للشركة . الوحيد فقط الذي يجب أن يكون بإمكانه بلوغ مجمع المعطيات في الكتابة . وعلى العكس ، فإن إستعمال مجمع المعطيات هو مفتوح لعدد غير مُحدَّد من الأشخاص (بدون تحفظ ، من الواضح لإجازات سماح معينة ، ليس بإمكان هؤلاء المستعملين العمل إلا بالطريقة المُنظَّـمة بها فيزيـائياً سجـلات مجمع المعـطيات ؛ وبدون أية علاقة فيما بينهم مع الأخرين . وإذا كان بإمكان كل مُستعمل أن يُعدِّل مضمون مجمع المعطيات ، فإن قيمة هذا الأخير ستصبح عديمة وبدون فائدة .

3.12 ـ السلاسل التأريخية اليومية

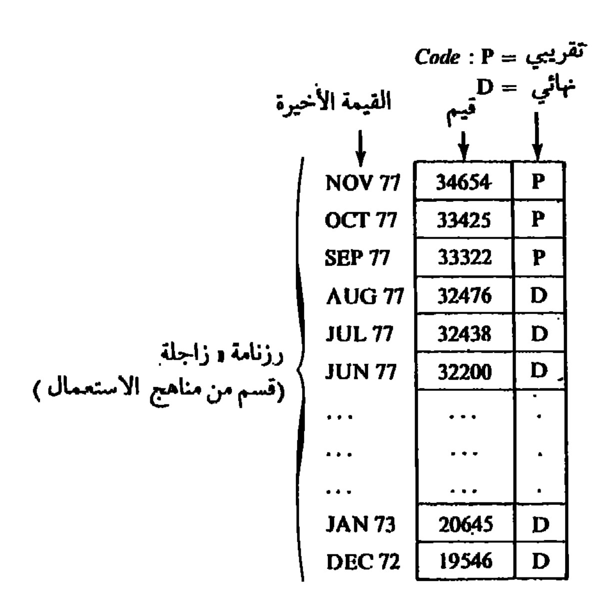
السلسلة اليومية هي جدول ببعد واحد أو بمؤشر واحد (إذا فضلنا تسميته : مُتجه) من القِيم الرقمية المتتالية المأخوذة من متحولة إقتصادية أو إدارية . مثلاً ، المتحولة « مبيع

المتوجة X ، بالعدد » سيجري قياسها شهراً بعد شهر . القيم المتتالية لهذه المتحولة تؤلف السلسلة الشهرية « مبيع المتوجة X ، في العدد » . وإذا حفظنا هذه المعطيات في نسخة زمنية من خمس سنوات ، سيكون عندنا 60 قيمة رقمية في السلسلة : نقول إذاً إن الدورة هي الشهر ، وإن طول السلسلة هو من 60 دورة (شكل 6) .

هناك فائدة كبرى في هذا التمثيل الداخلي للقيم الرقمية هو من نوع صحيح ، أو ثنائي صحيح .

هكذا ، يجب أن يكون هناك توافق بالنسق بين برامج الاستعمال من جهة ، وزجل (أو برامج) الإستيفاء اليومي وإنشاء مجمع المعطيات من جهة أخرى . هذه الأخيرة هي غالباً ما تكون مكتوبة بلغة COBOL ، أو بلغة متكيفة مع معالجة المعطيات ، بينما برامج الإستعمال ، غالباً هي بصفة علمية ، وتكون مكتوبة في لغة FORTRAN وفي نظام الوقت المقسَّم .

إذاً إستعمال النسق الثنائي الصافي هو الأكثر سهولة ، بوصف «COMPUTATIONAL» عناصر الجدول التي تؤلف السلسلة على أنها «COMPUTATIONAL» في لغة الكوبول ، أو بالتصريح «INTEGER» عن نفس الجدول في لغة FORTRAN ، مع الإنتباه للمشاكل حول طول الكلمات ، وبالأخص إذا كنا نعمل على حاسب بباية .



NOV 77	34654	P
OCT 77	33425	P
SEP 77	33322	P
AUG 77	32476	D
JUL 77	32438	D
JUN 77	32200	D
...	...	.
...	...	.
...	...	.
JAN 73	20645	D
DEC 72	19546	D

شكل 6 ـ القسم الرقمي من سلسلة شهرية

من البديهي إن تحديد طول السلسلة (اللائحة) يتم عند تصوُّر مجمع المعطيات ، حسب مهام الاستعمال المتوقعة في مادة الإحصاء . الخبرة العملية تبرهن على أن الحل الأمثل ، للوائح الشهرية ، هو طول من 60 دورة ، بينما اللوائح السنوية هي ذات معنى كاف مع 20 إلى 25 دورة . من الواضح ، إننا لا نستطيع الحصول على أكثر من القِيم ، في حالات اللوائح المنشأة من جديد . الخلايا المناسبة ستكون فارغة . فقط ، بعض المناهج الاحصائية لن تتمكن من العمل على هذه اللوائح القصيرة .

يجب أيضاً توقع تسجيلات لوصف اللوائح : عنوان ، مصدر ، مضمون خلايا ، مراصف أعداد أو قيم ، دوريات ، طول ، الخ . من الأسهل تجميع هذه التسجيلات الضرورية ، في قسم آخر من مجمع المعطيات .

نصل وبسرعة الى مفهوم تصور مجمع المعطيات حيث يؤدي جدول عام باللوائح الى بلوغ جميع مركبات هذه اللوائح : وصف ، قيم ، وإحتمالا ملاحظيات ، حماية ، مراقبة وإحصائيات إستعمال (شكل 7) . هذا الجدول يمكن أن يحتوي على كود من الأحرف لكل لائحة أو سلسلة .

التنظيم العام لهكذا مجمع للمعطيات هو شكل من التعميم لتقنية السجلات المتتالية والمؤشرة (I.S.A.M) . ومن الأنسب توقع مجموعة من السجلات المعكوسة، لعمليات التجميع المنطقية للّوائح حسب القطاعات ، أو حسب القوائم . مرة أخرى نقول ، إن التصوُّر العام لمجمع للمعطيات يتعلّق بالكامل بالاستعمال الداخلي لبنك المعطيات ، ولا يوجد أية قاعدة مشتركة في هذا المستوى .

وفي النهاية ، يمكن للائحة يوميات أن تحصل على مدة حياة محددة. هذه هي الحالة، مثلاً ، لمنتوجة لن تُصنع . ولكنها أيضاً حالة دليل تشغيل حيث التعريف يتغيَّر في أحد الأيام . يجب على النظام أن يأخذ بالحسبان هذه الحوادث .

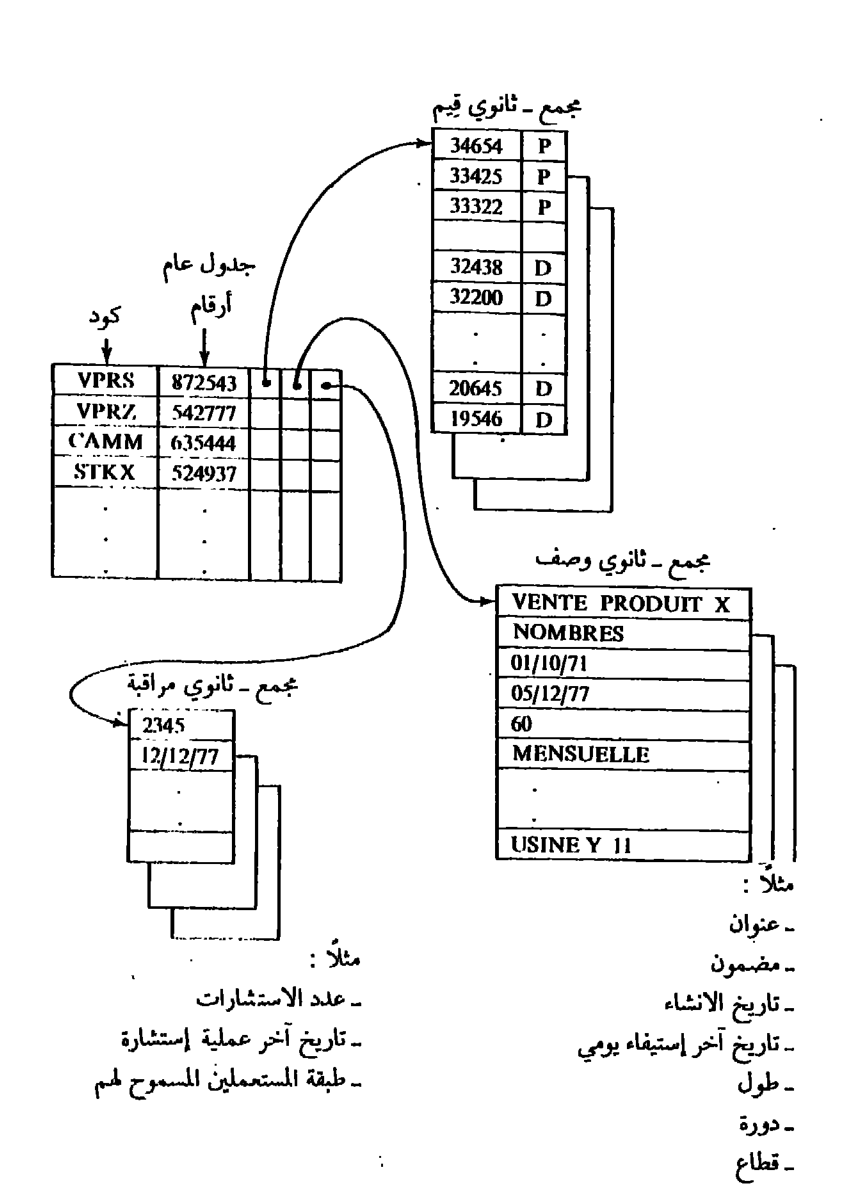
3.2 ـ إنشاء وإستيفاء يومي

3.21 ـ إنشاء أولي لمجمع المعطيات

الطريقة المؤكدة تقوم ، بعد تحديد تركيبة مجمع المعطيات والسجلات المُحضَّرة (« المنسَّقة formattés) ، على شحن متتال ، لائحة بعد لائحة .

هذا الثمن يتطلب برامج إنشاء النظام ـ الثانوي للاستيفاء اليومي للمجمع . ويُمكن أن يتم تنفيذه من خلال البطاقات ، الأشرطة المغناطيسية ، السجلات على الأقراص أو بواسطة الأدوات الطرفية التخاطبية .

فقط الحجم ، والإنشاء الأولي لمجمع المعطيات لا يختلف أبداً عن الإنشاء البسيط للوائح .



شكل 7 ـ التنظيم العام لمجمع المعطيات ، خاص ببنك معطيات إقتصادي

3.22 ـ إستيفاء يومي

الاستيفاء اليومي وكما رأينا ، هو المهمة الأكثر أهمية والأصعب لصيانة مجمع المعطيات

فهو يجب أن يكون مهمةمركز موحد للخدمة، وهو الطريقة الأفضل لتأمين تفاهم جيد في العمل . يؤمن مركز الخدمة هذا ، وبشكل خاص ، تركيز جميع الوثائق المعلوماتية

أو غير المناسبة لعمليات الاستيفاء اليومي وتتابعها . كما ويراقب حسن تنفيذ إجراءات الاستيفاء اليومي الأوتوماتيكية .

أغلب برامج الاستيفاء اليومي مكتوبة بلغة كوبول وتخرج عن معلوماتية الانتاج . بسبب معلوماتية الانتاج هذه ، يقوم بنك المعطيات الداخلي باستخراج جميع المعطيات التي سيجري تخزينها لاحقاً على شكل لوائح . نتحقق أيضاً من إن وجود معلوماتية جيدة للانتاج هو شرط ضروري لمعلوماتية إدارية جيدة . وعلى العكس ، فمن الصعب وضع نظام معلوماتي للإدارة في شركة معينة إذا كانت هذه الأخيرة غير مجهَّزة بأتمتة صحيحة للمعلومات وإنتاجها .

يحتوي النظام ـ الثانوي للاستيفاء اليومي وبالتحديد على مهام إنشاء لوائح ، إلغاء ، إضافة عناصر في سلسلة من التعديلات لواحد أو عدة عناصر ، تغيير قاعدة المرجع في حالة وجود سلاسل من الدلائل . . . الخ . وهو يحتوي أيضاً على جميع البرامج المطلوبة لتنقيح طبقات اللوائح ، المراقبة ، الخ .

وفي النهاية من المناسب الانتباه إلى مشاكل التجانس العامودي ، أو اليومي ، للوائح . هكذا فقد يحدث عادة أن يتغيَّر التعريف المحدَّد لمتحولة معينة خلال الوقت . هكذا مثلاً هو دليل الأسعار التفصيلية في بلد معيَّن (بنوك معطيات ماكرو ـ إقتصادية) : فهو يُحسب بطريقة معينة وحتى تاريخ معين ، بعدها تتغير طريقة الحساب . بعد ذلك لن يكون للسلسلة الوحيدة لهذا الدليل أي معنى للاستعمال . من المناسب هنا إغلاق اللائحة عند تاريخ التغيير في طريقة الحساب ، وإنشاء لائحة جديدة ، مختلفة عن السابقة .

3.3 ـ الاستعمال

لقد رأينا أن الأداة الصالحة لاستعمال بنك للمعطيات من هذا النوع هي « الوقت ـ المقسَّم »(*) . وعملياً ، يُفضَّل أن يكون بتصرفنا برامج لا يتطلب إستعمالها أي تعليم في البرمجة . تنقسم هذه البرامج إلى ثلاث طبقات : تحضير المعلومات ، الحسابات البسيطة على هذه المعطيات ، وفي النهاية الحسابات المركبة ، لطبيعة إحصائية .

3.31 ـ تحضير المعطيات

هي نقطة شديدة الأهمية . لأنه من غير المفيد الحصول على مجمع للمعطيات كامل على تشغيل الشركة إذا كانت المعلومات المستخرجة هي غير مقروءة أو غير مفهومة : من المهم أن تكون هذه المعلومات ، الضرورية لأخذ القرارات ، مفهومة بشكل واضح من قبل المُقرِّر . يوجد مشكلة تشكيل هذه المعلومات التي لا يجب أن يستخف بها المعلوماتيون ، تحت حجة أنه لا يقوم بطلب خوارزميات معقدة . لا يوجد شيء أكثر

(*) Time-sharing ـ وقت مقسم = Temps partagé : إستعمال تخاطبي لنظام ، من خلال أداة طرفية تخاطبية .

إزعاجاً ، عندما لا نكون إختصاصيين بالمعلوماتية ، من القيام بتفسير مطبوعة سريعة «Listing» ، أو إستشارة مخطوطة أمام وثيقة خارجة من أداة طرفية .

تحضير المعلومات يتم عادة على شكل جداول أو بشكل رسوم بيانية .

الجداول هي أكثر سهولة للانتاج والاخراج على أي أداة طرفية في الوقت ـ المقسَّم . يكفي فقط أن نتذكر إن الملاحظيات الواضحة وغير المختصرة ليست بكثيرة .

وفي الحد الأقصى ، يُمكن لبنك معطيات أن يصبح كافياً ، منذ أن يتم تجهيزه بنظام جيد لتنقيح الجداول . بعض الأدوات الطرفية هي قابلة لتنقيح رسوم بيانية ، بالتحكّم بالفراغات الأفقية والعامودية في جزء من الملم تقريباً . وإذا كان بتصرفنا هكذا عتاد ، فسيكون هناك فائدة كبرى من تجهيز الجداول ، الضرورية لمعرفة المعطيات الخام ، برسوم بيانية مناسبة ، أكثر وضوحاً وتعبيراً . لذا يعرض صانعو الأدوات الطرفية عادة ، مناهج للرسم : مكتوبة غالباً بلغة فورتران ، وهي سهلة التكييف مع أي حاسب .

3.32 ـ حسابات بسيطة

من الممكن توقع إضافة بعض عمليات الحساب إلى برامج التحضير الخام .
لذا ، فمن الممكن إضافة لوائح فيما بينها ، ضرب لائحة بعدد ، تجميع مختلف قيم اللائحة في ردحة معينة من الوقت ، إجراء حسابات لنسبة الزيادة من تاريخ معين إلى تاريخ آخر ، الخ .

3.33 ـ حسابات إحصائية مركبة

من الممكن أن توضع بتصرفنا مناهج أكثر تطوراً : تشديات (تمسيد) بواسطة وسائط متحركة ، أسية من مختلف الأنواع ، تضبيط مُتعدِّد الجذور ، طرق BOX وJenkins ، الخ .

في هذا المستوى ، يمكن أن تستعمل قوة الحاسب لوضع نماذج ، إجراء محاكاة ، فحص هذه الاستراتيجية أو تلك الخ . في هذه الحالات ، قد يكون مفيداً ربط المعطيات الداخلية للشركة بالمعطيات على المحيط ، ذات الطبيعة ماكرو ـ إقتصادية .

هذه الأخيرة يمكن أن تكون جاهزة بشكل معلوماتي ، بعض التنظيمات توزع لوائح من هذا النوع على أشرطة مغناطيسية ، أو بواسطة شبكات دولية للوقت المقسَّم .

هذه التقنيات ، المستعملة بشكل صحيح ، يُمكن أن تؤدي خدمات كبيرة : ولكن يجب أن لا ننسى أنّها لا تأخذ معناها إلا بالارتكاز على مجمع معطيات بإمكانية عمل كبيرة ، وإلا فكل هذا هو هراء .

4 ـ مفاهيم حديثة

4.1 ـ مجامع المعطيات العلائقية

هذا المفهوم ، يعود إلى CODD ، وتاريخه يعود إلى 1970 . ولا يوجد شيء مشترك مع توصيات CODASYL ، التي تعود تقريباً إلى نفس العصر .

أنطلق CODD من الملاحظات التالية : مفهوم السجل الكلاسيكي بنسق ثابت هو متقابل (مرتبط بعلاقة تقابلية أو نظام موحد في العلاقات) أو Isomorphe مع العلاقات الرياضية بدرجة n(n-aire) . إنطلاقاً من هنا ، فقد قام بتطوير نظرية مجامع للمعطيات تختلف تماماً عن مفاهيم CODASYL (SET ، MEMBRE ، مؤشرات ، الخ) .

مجمع المعطيات العلائقي هو مجموعة مشكلة من مجموعة علاقات من جميع الأنواع . يمكن لكل علاقة أن تتغيَّـر حسب الوقت . الوصف الكامل سيتطلب تشكيلاً رياضياً متطوراً : ويخرج عن إطار هذا الكتاب . ولأكثر دقة يُنصح القارىء بالعودة إلى المراجع .

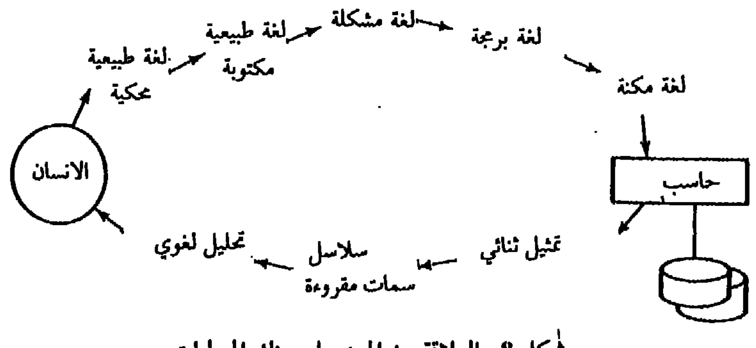
يكفي هنا أن نعرف إن مجامع المعطيات العلائقية تفتح خطاً لاستعمال سهل وفعّـال لمجامع المعطيات الكبيرة ، التي تستدعي بشكل خاص كثيراً من تداخل حساب المحمولات (predicats) لتركيب الأسئلة المتعددة . وتستعمل إختيارياً في إطار ما يدعى الآن « الأنظمة الخبيرة Systems experts) .

4.2 ـ إستعمال اللغة الطبيعية

الأداة الطرفية الأكثر طبيعية بالنسبة للانسان هي التلفون وليست الطابعة أو غيرها .

في السلسلة المزدوجة لتحويل المعلومات التي تضع المستعمل وبنك المعطيات في علاقة فيما بينهما (الإنسان والآلة) ، نفتقد حلقتين (زريدتين) : الفهم ، بواسطة نظام معلومات ، اللغة المحكية ، والتفسير الأوتوماتيكي للغة الطبيعية (شكل 8) .

في الاتجاه من الآلة نحو الإنسان ، يمكن أن نتكلم عن جميع المشاكل المحلولة :



شكل 8 ـ العلاقة بين المستعمل وبنك المعطيات

الحاسبات هي حالياً قادرة ، بواسطة أجهزة مُتخصِّصة (آجهزة تحليل أصوات) ، من الإجابة على التلفون .

الأصوات المرسلة ، الواضحة ، هي تركيبية بالكامل ، وغير مسجلة سابقاً .

في الإتجاه الآخر من الإنسان نحو الآلة ، يضع تكويد الإشارة الصوتية مشاكل متعددة . ولكن تفسير اللغة الطبيعية المكتوبة بدأ يعمل بشكل صحيح ، على الأقل في بعض الحالات : استعمال مجموعة ثانوية من اللغة الطبيعية المُخَصَّصة إلى بنك معطيات محدد .

5 بنوك المعطيات والمعلومات

في جميع الحقول ، العلمية ، التقنية ، الاقتصادية ، الحقوقية ، الوثائقية ، تحاول المعلومات أن تتخلى عن الورق ، الثقيل الوزن ، والباهظ الثمن ، لصالح نواقل المعلوماتية .

هذا الأخير يحمل قيمتين كبيرتين : قدرة الفرز على الحاسب ، إضافة إلى إمكانية التوزيع الانتقائي اللحظي حسب المعيار العالمي ، بسبب وجود شبكات المعلوماتية البعيدة .

هناك عدة بنوك للمعلومات موجودة الآن ، والتي توزع المعلومات ، على المشتركين بواسطة شبكات معلوماتية بعيدة خاصة وللجمهور . هذه الشبكات هي في الأصل MARK III ، TYMNET ، CYBERNET (تجارية ، USA ، بتوسيع عالمي) ، ARPA (للجمهور ، USA) ، EURONET (للجمهور ، أوروبا) ، TRANSPAC (للجمهور ، فرنسا) .

أما الخدم (Serveurs) ، فهي الحاسبات من جميع الأنواع ، الناقلة لبنوك المعلومات والمربوطة بالشبكات . وهناك شبكات عامة أو خاصة تدير العمل الضخم بصيانة بنوك المعطيات التي توزِّعها . وهي صاحبة المسؤولية والتحكُّم . هذه الشركات هي أمريكية في أغلبها .

وفي النهاية ، فإن سجلات بنوك المعطيات الأسمية تسمح بإمكانية قيادة ، تحكُّم ومراقبة عميقة ودقيقة في البلاد .

مراجع

كتب أساسية في أنظمة إدارة مجامع معطيات S.G.B.D

الوثائق التقنية حول S.G.B.D. ، الموضوعة بتصرفنا بواسطة موزعي هذه المناهج .
من الممكن أيضاً إستشارة الكتب التالية .

E. F. CODD, A Relational Model of Data for Large Shared Data, *Banks. Comm., ACM*, Vol. 13, N° 5 (June 1970).

CODASYL, Report of the Codasyl Data Base Task Group, *ACM* (New York, 1971).

C. DELOBEL et M. ADIBA, *Bases de données et systèmes relationnels*, Dunod, 1982.

A. FLORY, *Bases de données, conception et réalisation*, Economica (Paris, 1982).

Ch. BERTHET, A. MESGUICH et B. NORMIER, *The problems of applying informations systems in organizations : communication between man and computer*, Institut Européen de Recherches et d'Etudes Supérieures en Management, Bruxelles (1975).

D. MARTIN, *Bases de données : méthodes pratiques sur maxi et mini-ordinateurs*, Dunod-Informatique, Paris, 1981.

Répertoires de Banques de données :

DIRECTORY OF ONLINE DATABASES, Vol. 3, n° 3, Spring 1982, Cuadra associates, Santa-Monica, 1982.

DATA BASES IN EUROPE 1982, Répertoire des Banques et des serveurs européens connectés au réseau Euronet-Diane, C.C.E. Luxembourg, 1982.

BASES ET BANQUES DE DONNÉES ACCESSIBLES EN CONVERSATIONNEL EN FRANCE, Association Nationale de la Recherche Technique (ANRT), Paris, 1981.

GUIDE THÉMATIQUE DES BANQUES DE DONNÉES FRANÇAISES, Centre d'information des Banques de données et du Vidéotex (CIBDV), Paris, 1982.

RÉPERTOIRE DES PRODUCTEURS DE BASES ET BANQUES DE DONNÉES, Groupement Français des Producteurs de Bases et Banques de données (GFPBD), Paris, 1982.

INTERACTIVE COMPUTING DIRECTORIES, Association of Time Sharing Users (ATSU), USA (annuel).

Organismes assurant la promotion des systèmes d'information automatisés :

MIDIST (Mission interministérielle de développement de l'information scientifique et technique); 280, boulevard St-Germain, 75007 Paris. Tél. 550.32.50.

DBMIST, ministère de l'Education nationale; Direction des bibliothèques, des musées et de l'information scientifique et technique, 3, boulevard Pasteur, Paris 75015. Tél. 539.25.75.

ANRT (Association nationale de la recherche technique), 101, avenue Raymond-Poincaré, 75116 Paris. Tél. 501.72.27.

CNIC (Centre national pour l'information chimique), 26, rue Boyer, 75020 Paris. Tél. 797.29.29.

CDST (Centre de documentation scientifique et technique du CNRS), 26, rue Boyer, 75020 Paris. Tél. 358.35.59.

GFPBD (Groupement français des producteurs de banques de données), 103, rue de Lille, 75007 Paris. Tél. 550.32.22.

ADBS (Association des documentalistes bibliothécaires spécialisés), 5, avenue Franco-Russe, 75007 Paris. Tél. 555.55.16.

CIBDV (Centre d'information des banques de données et du vidéotex), 11, rue du Marché-St-Honoré, 75001 Paris. Tél. 261.45.17.

ATSU (Association of Time-Sharing Users), 75 Manhattan Drive, Boulder Co, USA.

Législation :

LOI N° 78-17 DU 6 JANVIER 1978, *relative à l'informatique, aux fichiers et aux libertés*, Journal Officiel du 7 janvier 1978, pp. 227 à 231.
+ *Décret* 78-774 du 17-07-78
+ *Décret* 78-1223 du 28-12-78
+ *Décret* 79-421 du 30-05-79
+ *Décret* 79-1160 du 28-12-79.

RAPPORTS DE LA COMMISSION NATIONALE INFORMATIQUE ET LIBERTÉS (CNIL), à la Documentation Française (Paris).

CNIL, 21, rue Saint-Guillaume, 75007 Paris, Tél. 544.40.65.

الفصل الخامس

لغات البرمجة

تُدعى لغات للبرمجة ، اللغات الخارجية ، أو اللغات المتطورة ، على عكس مختلف « لغات المكنة » (أو اللغات بمستوى I) ولغات التأويل (اللغات من مستوى II) .

ولقد أصبحت لغات البرمجة هذه (لغات بمستوى III) ، الوسيلة الرئيسية لاتصال الخوارزم بالحاسب . وكما إن بنيتها قريبة من بنية هذه الأخيرة فهي تدعى غالباً لغات خوارزمية (تراتبية) .

لكل من اللغات ، عدد من السيئات والحسنات . وبشكل خاص ، فهي تحتاج إلى مدة للتصريف . ولكن وبما أنه من الأسرع كتابة برنامج بلغة خوارزمية أكثر من لغة المؤول ، وبما إن ثمن سعة الحاسب هو في تناقص بينما ساعة عمل المهندس هي في إزدياد ، فإن هذه السيئة (مدة التصريف) بدأت بالزوال .

نسبياً ، هذه اللغات هي سهلة للفهم والتعلّم ، وهي أكثر مرونة من لغات المكنة ، وتعديل البرنامج المكتوب بلغة خوارزمية يتم بسرعة أكبر من أي تعديل بلغة المؤول (assembler) .

يوجد حالياً حوالي مئة من لغات البرمجة المختلفة ، بعضها ليس سوى « ديالكتيك » للأخرى . نصف هذه اللغات مصنوع لاجراء تطبيقات مُتخصّصة ، ومثلاً للمحاكاة (GPSS ، SIMSCRIPT ، الخ) ، وهناك حوالي 15 لغة كثيرة الاستعمال من بينها :

للتطبيقات العلمية الرقمية	FORTRAN ALGOL
كاللغات في الوقت المقسّم	BASIC APL
كلغة للإدارة	COBOL
لمعالجة اللوائح (*)	LISP
لمعالجة سلاسل السمات	SNOBOL
لجميع الأعمال التطبيقية الممكنة	PL/1

(*) إستعمال LISP عرف نجاحاً كبيراً حالياً : وهو الواسطة الأساسية للأنظمة الكبيرة «Systèmes experts» .

1 ـ فورتران FORTRAN

هذه اللغة هي الأكثر شعبيةً . وتاريخها يعود إلى بداية ظهور المكنات . وهي كثيرة الفائدة في التطبيقات العلمية الرقمية وتبقى جاهزة على أغلب الحاسبات .

ففي 10 تشرين أول سنة 1954 ، نشرت مجموعة الدراسات والأبحاث في الشركة العالمية للمكنات IBM دراسة هي :

«Preliminary Report. Specification for the IBM Mathematical FORmula TRANslation, FORTRAN».

لم يتم تقبُّل هذه اللغة بسهولة . وكان البعض يحتج على أن عمل المُصرِّف هو أقل جودة من أفضل المبرمجين بلغة المكنة . إضافة لذلك فإن هذه الصيغة للفورتران كانت قليلة الاستقلالية عن المكنة .

هكذا ، وفي سنة 1958 ، جرى نشر صيغة جديدة : FORTRAN II . وكان الجديد فيها هو إمكانية إستعمال برامج ثانوية («SUBROUTINE» ، و«FUNCTION») .

وحتى سنة 1961 ، التاريخ الذي جرى فيه إعتماد مصمِّمين آخرين غير IBM ، للغة فورتران على مكناتهم (Remington Rand UNIVAC) ، جرى تطوير عدد كبير من مصرفات FORTRAN ، وكان لكل منها ميزاته الخاصة ، المختلفة من لغة إلى أخرى . ولمعالجة هذه الحالة ، جرى في آذار سنة 1961 ، تشكيل لجنة للمستعملين تدعى «SHARE FORTRAN commitee» ، لوضع صيغة جديدة للغة فورتران ، حيث FORTRAN II لم يكن بأي شكل مجموعة ـ ثانوية . وفي نفس الوقت ، جرى إنشاد FORTRAN III في IBM، وكان يحتوي كجديد على مواصفات وتحديدات بولية (boolean) .

وفي سنة 1962 ، نشرت لجنة «STARE FORTRAN» صيغة جديدة للغة فورتران تُعرف الآن بالصيغة FORTRAN IV .

ولقد شكلت اللجنة الأميركية للنماذج «Americain standard organisation» لجنة X3.4.3 لوضع معايير للغة FORTRAN . ولقد جرى تحديد لغتين هما : FORTRAN الكامل ، وأسس FORTRAN المعتمدة هي اللغة الأولى بواسطة معهد النماذج في الولايات المتحدة (USA) .

وفي سنة 1963 بدأ جميع المصمّمين يعتمدون لغة FORTRAN .

المميزات التقنية للغة FORTRAN

هناك ثلاثة مفاهيم جعلت من هذه اللغة وسيلة فعّالة للحساب العلمي :

ـ ترجمة الصِيغ العلمية (Formula-translation) ، مثلاً الصيغ التالية :

$$y = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^5}{24}$$

$$z = 1 + \sin(x + 1)$$

$$\delta = \sqrt{b^2 - 4ac}$$

تكتب بلغة فورتران :

```
Y = 1. + X + X**2/2. + X**3/6.
         + X**5/24.
Z = 1. + SIN (X + 1.)
DELTA = SQRT (B**2 - 4.*A*C)
```

كتابة هذه الصيغ بلغة فورتران هي قريبة من الكتابة الرياضية الكلاسيكية :

الحلقة DO (تكرار متتالي لسلسلة تعليمات) . هذا التحديد هو واحد من التحديدات الأكثر فائدة . وهو يسمح ، بين الأشياء الأخرى بحساب التكامل ، ومعالجة المصفوفات ، الخ .

ـ البرامج ـ الثانوية (من نوع «FUNCTION» أو «SUBROUTINE») . هذه البرامج الثانوية هي مستعملة بواسطة برامج FORTRAN بشكل مستقل ويُمكن أن تُشكِّل مكتبات .

لغة فورتران هي سهلة للفهم . والمصرفات هي فعالة ،وتعمل بسرعة وتولد أكواد فعالة . ولكنها تعاني من ولادتها باكراً ، في عصر لم يكن يتوقع فيه الإمكانيات الحالية للحاسبات . وإذا كانت هذه اللغة هي جيدة في الحساب الرقمي ، فلا تتمتع بمواصفات وتحديدات في معالجة سلاسل السمات (إلا بشكل برامج ثانوية غير ملائمة للاستعمال) . ما عدا ذلك ، فهي لا تحتوي على تراتبية (عشيرية) في المعطيات .

لذا يبقى فقط إستعمالها في حقلها الخاص ، الحساب العلمي الرقمي ، وهي تواصل تقديم الخدمات الكبيرة ، على جميع المكنات ، حتى الصغيرة منها ، ويجب أن تكون قسماً من المعلومات الأساسية لكل شخص يرغب باستعمال الحاسبات .

2 ـ الكوبول COBOL

تُطبق لغات الإدارة على المسائل ذات السجلات الكبيرة والتي يجب العمل عليها مباشرة (إستيفاء يومي) . المعالجات والحسابات هي سهلة بشكل عام . وبما إن السجلات الكبيرة لا يمكن أن تُحفظ في الذاكرة المركزية ، لذا يجب أن تكون لغات الادارة مُجهَّزة بوسائل إدخال ـ إخراج متطورة .

المفهوم الثاني الأساسي لهذه اللغات هي في كونها الأقرب إلى اللغات الطبيعية ، وهي مؤلفة من كلمات وأفعال طبيعية لوصف المعطيات والعمليات المتعلقة بها .

تابعاتها

فيما بعد ظهر عدد كبير من اللغات المُخصَّصة للإدارة وبسرعة . وإستعمال الحاسبات كان مهماً في الإدارة (أكثر من 70%) . ومن جهة أخرى ، نلاحظ ، إن تطوُّر وتعديل أغلب برامج الإدارة مثلاً ، برنامج للمدفوعات ، برنامج المحافظة على المخزون (مخزون أحد المستودعات) . . . الخ ما زال مستمراً . هذه التعديلات هي سهلة الأجراء في اللغات الخوارزمية أكبر منها في لغة المكنة . وفي النهاية ، يجب أن يكون بإمكاننا وبسهولة ربط أحد البرامج ، وفهم ما يعمله ويقوم به بسرعة : من هنا الحاجة ، وبالنسبة للغة البرمجة ، إلى أن تكون أكثر قرباً من اللغة الطبيعية .

تقريباً وفي نفس العصر ، قام مختلف المُصممين بتطوير لغات الإدارة على هذه المفاهيم .

- FLOW-MATIC (من REMINGTON Rand UNIVAC ، 1955) وكانت مُقدَّمة كلغة قريبة من اللغة الإنكليزية ، ومجهَّزة بقسم لوصف المعطيات ، مستقل عن إجراءات المعالجة .
- AIMACO (Air Force Air MAterial COmmand) وجرت صناعتها في نفس الوقت على IBM 705 وعلى UNIVAC 1105 .
- COMTRAN (COMmercial TRANslator, IBM, 1958) إنطلقت من نفس مفهوم FLOW-MATIC ، وكانت مُجهَّزة بإمكانيات خاصة لمعالجة الصيغ الرياضية (التقنية في فورتران) ، وتحتوي على التحديدات المهمة «IF ... THEN..» والتي أدت إلى إدخال مفهوم التراتبية في المعطيات
- FACT (Fully Automatic Compiling Technique, Minneapolis- Honey well Regulator Co Data division, 1959)

  لم يتم تنفيذها على مكنة أخرى غير المكنة 800 ، بسبب لغة COBOL وضغوطات الإدارة الأمريكية بخصوص هذه اللغة .

لغة كوبول

في سنة 1959 ، أثار (Charles. A. Philips) ، من قسم الدفاع في البنتاغون إجتماعاً للمُصمِّمين ، المستعملين ، الجامعيين وأعضاء الإدارة الأمريكية بهدف إحداث لغة إدارة مشتركة . ولقد جرى تحديد تصوُّر من ثلاثة لجان (Short, Intermediate et long Range) للعمل . كانت اللجنة الأولى «Short Kange committee» تحتوي على

الشركات Burroughs ، IBM ، Minneapolis-Hony well ، RCA ، Rem—ngton- Rand Div. of sperry rand (Univac) ، Sylvania Electric products ، والوكالات الحكومية Air Material Command USAF وDavid Taylor Model Basin ، من De- partement of the Navy .

إجتمعت اللجنة لأول مرة في 23 حزيران 1959 ، ووضعت ملاحظاتها على اللغة «Common Business oriented langage» ، أو COBOL في نهاية 1959 .

أما الشركات Control Data Corporation ، General Electric ، NCR ، وأخرى فقد ارتبطت بهم سنة 1960 .

ولقد جرى مُعايرة لغة COBOL في سنة 1963 (USAI X 3.3.3) ، وتم إنشاء صيغة تُدعى حالياً COBOL-65 . وقد إشترك في تصميمها المنظومة (European ECMA Computer Monufacturer's Association) . ولقد جرى تحديد اللغة كنموذجية (USA Standard X 3.23.1968) في آب سنة 1968 . وكانت جميع المكنات المسلّمة الى الدولة الأمريكية تُسلّم مع مصرِّف COBOL ، إلا في بعض الحالات الشاذة القليلة (المكنات المصنوعة للعمل في الوقت الفصلي ، مثلاً) .

المميزات التقنية للغة كوبول

اللغة ليست عامة ولا مختصرة ، ولكنها قريبة من اللغة الانكليزية . تقسيم المعطيات هو مرهق للكتابة ، وإسهاب قسم المعالجة يعطيها صفة التكثيف .

لا تحتوي لغة COBOL على الدالات (في المعنى «FUNCTION» للغة FORTRAN) . والعدد الأقصى للدلائل هو ثلاثة في الجداول وهي موضع تصريح خاص مرهق في قسم المعطيات .

وعلى العكس ، يُمكن للمعطيات أن تتمثّل في البنى التراتبية . إضافة لذلك ، يوجد فعل «SORT» مفيد بشكل خاص إذا كنا نرغب بإجراء أي ترتيب للمعطيات .

لغة الكوبول هي غير لائقة بمعالجة اللوائح ، كما وتُعالج بشكل سيىء سلاسل السمات . وهذا هو سيىء بالنسبة لأي لغة إدارة . وعلى العكس ، فإن تنفيذ التعليمات الشرطية هو سهل وجيّد .

مثلاً :

```
IF A IS GREATER THAN B GO TO
                            PARAGRAPH-3.
```

وتعليمتها PERFORM ، غير الموجودة في أي لغة أخرى، هي من الوسائط المهمة .

ولقد حملت لغة COBOL إسهاماً كبيراً في تطوير المعلومات المناسبة للملاقي (Divi- sion Environnement) .

وفي النهاية ، فإن إمكانيات تطوير ومعالجة السجلات هي جيدة . لغة البرمجة هذه هي شديدة الفعالية في مسائل معالجة السجلات الكبيرة ، وفي الحالة التي تكون فيها معالجة المعطيات سهلة .

لذا فإن لغة كوبول هي وسيط إدارة مفيد ، ولكنها محدودة . وسيكون هناك فائدة من تبديلها بلغة أخرى للإدارة ، جديدة ومتكيفة بشكل أفضل مع تقنيات المعلوماتية حالياً .

BASIC - 3

صنع هذه اللغة الأساتذة Kemeny وKurtz ، بمساعدة جنرال الكتريك ، في دارموث كوليدج (Darmouth College) في الولايات المتحدة ، وهي جامعة صغيرة مشهورة في حقل المعلوماتية . وكانت مهمتها تقديم لغة جبرية بسيطة للطلاب ، تسمح لهم ببلوغ المكنة في الوقت المقسّم (time sharing) :. اللغة BASIC (Beginners All pur- pose symbolic Instruction code) إشتغلت للمرة الأولى في سنة 1964 على المكنة GE 225 . وهي لغة قريبة من لغة فورتران ، حيث تستعمل نفس العناصر الأساسية مع تبسيطها . وهي مرتبطة بالوقت المقسّم ، حيث أطلقت التطوير الداخلي . . لهذا فكل تعليمة BASIC تبدأ برقم للسطر يسمح بمعالجة بسيطة لبرنامج المصدر ، وبتسريع العمل .

ومن البديهي أن لا يكون لهذه اللغة الحقل التطبيقي الخاص ، ولا السهولة في إدارة وتنظيم السجلات كما في لغة COBOL . وليس لها قدرة وقوة لغة FORTRAN ، ولا إمكانيات لغة PL/ 1 . ولكنها لغة جيدة للبدء بتعليم البرمجة (وعلى الأخص في الصيغ الموسّعة) وتقدم خدمات جُلى في هذا الحقل . وتطورها الحالي هو مرتبط بالتطور السريع للمعلوماتية الشخصية : نجد بازيك على جميع مكنات الميكروكومبيوتر . في فرنسا ، وللتطبيقات التعليمية ، عرّف J. Hebenstreit في سنة 1971 ، بطلب من وزارة التربية الوطنية ، لغة قريبة من بازيك تدعى LSE (Langage Symbolique d'Enseignement) . تستعمل اللغة LSE ألفباء فرنسية ، وهي سهلة للتعليم ، بدون أي توضيح : وتسمح ببرمجة عدد كبير من التطبيقات بتعقيد مختلف ، تبدأ من الحساب العلمي الكلاسيكي إلى البرامج المساعدة للتعليم في مختلف الحقول . وكلغة بازيك ، فهي صالحة للاستعمال على جميع الحاسبات الصغيرة(*) .

(*) لنحن واضحين يجب أن نشير إلى أن أول لغة علمية تخاطبية هي فرنسية : اللغة PAF (programmation auto- matique des formules) على CAB 500 في سنة 1958 .

4 ـ اللغات المتعددة الاستعمال

في سنة 1959 ، أنشأت الشركة : System Development Corporation du stra- tégie Air Command Control System. تـوسيعـاً للغـة (Internal Algebric IAL language) يُدعى JOVIAL ، بواسطة مجموعة كلمات عـلى إسم الشخص المهتم بها (Jules (Schwartz) own Version of the international Algebric langauge) . جرى تنفيذ هذه اللغة على عدة مكنات ، ولكنها تفتقد مع ذلك إلى المصرِّفات . ولكن لغة البرمجة المتعددة الاستعمال والأكثر أهمية هي لغة PL / 1 .

لغة البرمجة PL / 1

وبسبب عدم كفاية لغة فورتران في معالجة سلاسل السمات ، ومعالجة السجلات والتفاعل مع الأدوات الحديثة ، شكلت اللجنة «SHARE FORTRAN Committe» وشركة IBM ، في سنة 1964 ، لجنة لتطوير اللغات «Advanced language Develope- ment committe» درست اللغـات ALGOL ، COBOL ، JOVIAL كي تحاول أن تقوم بإجراء تحليل لها . ولقد أثار تقريرها الأول سنة 1964 ردات فعل مختلفة . البعض كان ممتازاً ، لجهة المفهوم التحليلي لهذه اللغات التي كانت تُجمِّع المـواصفات الأفضـل للغات FORTRAN ، ALGOL ، COBOL ، وأخرى سيئة : وكان يُـطلب معرفـة الفائدة من هكذا خليط .

ولقد حصلت اللغة الجديدة على الإسم (New Programming Language) NPL وبعد ذلك جرى إجراء عدة تعديلات عليها ، ومن ثم تقديم تقريـر ثـان عنهـا في نهاية 1964 ، وإنتهى باختيار الاسم PL/ 1 (Programming language Number 1) .

بينما كانت صناعة الحاسبات تهتم عن قرب بالموضوع ، كانت صناعة المصرِّفات تنطلق في Hursley في بريطانيا ، وفي مختبرات IBM بينما كانت اللغة تُطوَّر في الولايات المتحدة .

المصرِّف الأول (PL/ 1-F-360, IBM) كان جاهزاً في آب سنة 1906 . خلال هذا الوقت ، كان بعض المُصممين يهتمون بلغة PL/ 1 ، وينشئون فرق عمل وأبحاث ويطرحون مشكلة تكييف اللغة على مكناتهم .

الصيغ الأولى للمصرفات لم تقدم أي نجاح كامل . وبسبب تعقيد اللغـة ، فإن المصرفات كانت تعمل ببطء وتنتج أكواداً قليلة الفعالية .

وكان هناك بعض المستعملين الذي حكموا بشكل غير كاف على PL/ 1 . ولكن لا يجب الخلط بين اللغة ومصرِّف هذه اللغة . اللغة يُمكن أن تكون مهمة بينمـا قد يكـون المُصرِّف مخيباً للآمال .

ولكن الصيغة الرابعة للمصرِّف PL / 1- F- 360 (Hursly) كانت جيدة .

التصريف يتم بسرعة ، ويأخذ مكاناً صغيراً في الذاكرة ، ويولِّد كوداً قوياً وفعّالاً ، مع إعتبار الإمكانيات الكبيرة لمعالجة المعطيات التي تقدمها هذه اللغة .

ولكن التوافق مع لغة فورتران ليس كاملاً (بلوغ إلى نصف كلمة في النسق الثنائي الثابت) . الصيغة الخامسة لنفس المصرِّف تطمر هذه الفجوة في نفس الوقت الذي تضيف فيه مواصفات جديدة مهمة في معالجة سلاسل السمات وتزيد من فعالية الكود المُولَّد ، مما يجعل لغة PL/ 1 تقريباً بنفس سرعة فورتران وكوبول .

المصرِّفات الحالية هي فعالة بشكل ملحوظ ، كما بالنسبة للبرمجة كذلك بالنسبة لأفضلية الكود المُولَّد ، والتصورات المناسبة لقيمة PL/ 1 ، والصالحة في سنة 1966 ، ليست صالحة اليوم ، وعلى الأخص إذا أخذنا بالحسبان الوقت الذي نربحه اليوم من وضع البرامج في العمل .

فقط لغة PL/ 1 تسمح حالياً بمعالجة صحيحة للجداول والتركيبات ، وهذا غير ممكن في لغة FORTRAN وعلى الأخص بلغة COBOL .

تحتوي لغة PL/ 1 على جميع تسهيلات لغة فورتران في الحساب ، وتسمح إضافة لذلك بمعالجة أبسط للمصفوفات والجداول .

وهي تسمح أيضاً بمعالجة بسيطة للوائح بواسطة نظام التحولات المُؤشرة («BASED») وتبقى اللغة الوحيدة الخوارزمية المتكيفة مع معالجة السلاسل .

بواسطة نظام الماكرو تعليمات ، يمكن أن يتوسع على نفسه . وبسبب تركيبتها الزجلية والتراتبية فهي تصبح اللغة الأفضل للتربية والتعليم ، فلها تركيبة ALGOL ، مزودة بقدر وإجراءات .

وهي لغة جيدة للإدارة كلغة كوبول ، حيث تستعمل جميع التحديدات في معالجة السجلات مع تبسيطها

الآن ، ومنذ بداية مشروع PL/ 1 فلقد كان لها تأثير كبير على صناعة الحاسبات . وبإمكانها تقنياً أن تستبدل اللغات الأساسية الموجودة FORTRAN ، ALGOL ، COBOL .

ويعرض المُصمّمون الأساسيون اليوم مصرِّفات PL/ 1 فعالة . وهذا هو الحال مع IBM وكذلك مع BULL وUNIVAC أيضاً .

ADA _ 5

أمام الثمن الزائد في صيانة البرامج ، أطلقت وزارة الدفاع في الولايات المتحدة

(القسم DOD) في سنة 1975 نداءاً إلى تطوير لغة مؤكدة ، مقروءة ، تُسهّل البرمجة والصيانة .

وفي سنة 1977 جرى تسليم حوالي 17 عرضاً إلى وزارة الدفاع ، وجرى الاحتفاظ بأربعة منها (من Intermetrics ، Softech ، SRI International ، CII-Honeywell-Bull) . وفي النهاية جرى إختيار الأخيرة . هذه اللغة الجديدة المشتركة ناتجة عن عروض مشتركة،ولكنها بشكل أساسي من إختراع CII Honeywell- Bull ، ساعده بها Henry ledgard وآخرون كثر. ولقد جرى تسميتها ADA ، على شرف الكونتس Augusta Ada Lovelace ، إبنة الشاعر الإنكليزي Lord Byron وصديقة شارل بابدج (Charles Babbage) ، حيث إرتبطت بكتاباته . « إذا كان بابدج أباً للتكنولوجيا الحديثة في الحاسبات ، يقول هنري لادكار (Henri Ledgard) ، فإن Ada Lovelace هي (الإبنة الكبيرة للكومبيوتر) Great lady of Computers ، ويمكن أن تعتبر كزوجته العاقلة » .

ومن الباكر أن نقول إن هذه اللغة ، الواسعة الانتشار ، سيكون لها مستقبل باهر : لكن هذا الا مل محتمل وذلك بسبب صفاتها . إضافة إلى أن الحاجة إلى الحصول على معايير في مادة البرمجة أصبح ضرورياً و ADA تستفيد من الخبرة في المفاهيم الحديثة في البرمجة ، ويمكن أن تتلاءم كلغة عامة متعددة الاستعمال .

مراجع

كتب أساسية :

J. ARSAC, *La construction de programmes structurés*, Dunod-Informatique, Paris, Nouveau tirage 1981.

J. ARSAC, *Premières leçons de programmation*, Cédic-Nathan, Paris, 1980.

H. F. LEDGARD, *Proverbes de programmation*, traduits et annotés par J. ARSAC, Dunod-Informatique, Paris, Nouveau tirage 1981.

Pour le langage ADA :

H. F. LEDGARD et RAPPORT ADA, *ADA, an Introduction*; *ADA Reference Manual* (*July* 1980), Springer Verlag (New York, 1981).

D. LE VERRAND, *Le langage ADA, manuel d'évaluation*, Dunod, 1982.



الفصــل الســـادس

FORTRAN

1 ـ مواضيع اللغة

1.1 ـ مجموعة السمات

تستعمل لغة فورتران مجموعة من 48سمة : يُستعمل البياض (blanc)أو الفراغ لفصل مختلف عناصر التعليمة ، كي نحصل على قراءة أفضل . هكذا تهمل الفراغات بواسطة المصرِّفات ، إلاّ عندما تدخل إلى ثابتة هولورايت (أو ثابتة رمزية مؤلفة من سلسلة من السمات) . من الممكن أن نُدخل الفراغات إلى الكلمات أو لا ندخلها ، وإذا استعملناها فباستطاعتنا إدخال ما نريد منها .

لهذا فالكتابات التالية :

```
GO TO 1543
GOTO1543
```

هي متعادلة ، ولكن الشكل الأول ، الأوضح ، يُفضَّل عن الآخر . الأحرف : وتدعى أيضاً سمات أبجدية ، وهي الأحرف الستة وعشرين من الألفباء ، من A إلى Z . وتتمثل عادة في الأشكال الكبيرة (Majuscule)، ولكن هذا لا يمنع بعض المصرِّفات من استقبال الأحرف الصغيرة وتحويلها إلى كبيرة .

في فرنسا ، يُمثُّل الحرف O مع قضيب Ø ، مثلاً ØSCAR ، وفي أميركا الاتفاق هو العكس ، حيث يمثل الصفر مع قضيب Ø ، مثلاً : 15ØØ.25 .

ولكن هذا ، لا يعني أي تفريق في شكل الحرف O والصفر في أغلب الطابعات والأدوات الطرفية

وفي هذا الكتاب يُمثُّل الصفر على الشكل التالي Ø

الأرقام هي الأرقام العربية العشرة ، من 0 إلى 9 .

السمات الخاصة : هي السمات التالية :

+ زائد
– ناقص
* نجمة «Star»
/ قضيب قسمة «Slash»
= يساوي
. نقطة
, فاصلة
(هلال مفتوح أو يسار .
) هلال مغلق أو يمين .
' أبوستروف «Guote»
$ دولار

يرجى من القارىء مراجعة التمثيل الداخلي لهذه السمات في الملحق A . وفي الكود ASCII وEBCDIC .

1.2 ـ الثوابت

1.21 ـ الثوابت الجبرية

تعرف لغة فورتران نوعين من التمثيل الرقمي : النوع صحيح (entier) (ثنائي ثابت) والنوع مُتحرِّك (جزء عشري وأس) .

أ ـ الثوابت الصحيحة

وتُمثَّل بواسطة أعداد صحيحة عشرية ، إيجابية أو سلبية . القِيم القصوى تتعلَّق بمميزات المكنة المستعملة . وإذا كانت n هي الطول الثنائي للكلمة ، فإن أكبر قيمة ممكنة للعدد الصحيح هي : $2^{n-1} - 1$

مثلاً ، إذا كان n = 32 (مكنة بالبايتات) ، فهذه القيمة القصوى تعادل 2147483 647 .

أمثلة على الثوابت الصحيحة .

```
      12
   - 257
       0
+  126278
   - 8334
```

إستعمال الإشارة + ، في حالة الأعداد الصحيحة الإيجابية ، هو دائماً إختياري . إضافة لذلك ، فواحد أو عدة بياضات يمكن أن تفصل الاشارات عندما تكون موجودة في العدد .

ب ـ الثوابت بفاصلة متحركة .

تكتب هذه الثوابت بطريقتين مختلفتين

الشكل الأول

```
  18.53
- 257.01
   1.
 -12.
```

النقطة العشرية هي إلزامية ، حتى ولو كان العدد صحيحاً .

الشكل الثاني :

```
  0.1853E+02
- 0.25701E+03
  1.0E00
- 120.0E-01
```

E تعني « قوة عشرة أو أس » . الأس هو عدد صحيح يتألف من رقمين على الأكثر ، إيجابي ، سلبي أو 0 . الجزء العشري يكتب كتابته متحركة وبالشكل الأول .

ج ـ الثوابت بفاصلة متحركة وبدقة مزدوجة

من الممكن زيادة دقة الأعداد الحقيقية . والكتابة ليست مختلفة .

الشكل الأول :

```
  2.718281828
- 15.0023580132
```

الشكل الأول هو شبيه بالشكل الأول السابق ، بعدد الأرقام ذات الدلالة . النقطة العشرية هي دائماً إلزامية .

الشكل الثاني :

```
  2.718281828 D 0
- 1.50023580132 D + 01
```

الكتابة هي شبيهة بالشكل الثاني السابق ، ولكن الحرف D هو المستعمل بدلاً من الحرف E . وفي هذه الحالة أيضاً الأس (القوة عشرة) هو عدد صحيح يتألف من رقمين على الأكثر ، إيجابي ، سلبي أو صفر .

1.22 ـ الثوابت المنطقية .

الثابتة المنطقية لا يُمكن أن تكون إلا « حقيقة = true = vrai » أو « غلط faux I = FALSE » .

```
« vrai »  .TRUE.
« faux »  .FALSE.
```

ملاحظة : النقاط هي إلزامية .

1.23 ـ الثوابت الأبجعددية

وهي عبارة عن سلاسل السمات المختلفة . وحسب نوع المصرِّف يُمكن أن تُحصر بطريقتين مختلفتين : باتباعها بالمعرِّف nH ، حيث n هي طول سلسلة السمات ، ويكون صحيح وإيجابي ، أو باتباعها بواسطة أبوستروف (' ') .

أمثلة :

```
'ABCDEFGH'
'A01+2-/3AZ'
'JEAN-PIERRE'
'ALPHA BETA'
10HABCD1234YZ
2HbA
```

(b تمثِّل الفراغ) .

ملاحظة : جرى إختيار H بواسطة مخترعي FORTRAN على شرف Dr Holoreith ، مخترع البطاقات المثقوبة في شركةIBM .

1.24 ـ الثوابت المركبة

يُكتب القسمان ، الصحيح والوهمي من العدد المركب ، بالتعبير الحقيقي ، مفصولين بواسطة فاصلة أو بين أهِلّة .

أمثلة :

(12.544,24.012)	قسم صحيح	= 12.544
	قسم وهمي	= 24.012
(0.1,-27.33)	قسم صحيح	= 0.1
	قسم وهمي	= 27.33
(0.,0.)	قسم صحيح	

1.3 ـ المتحولات

1.31 ـ قواعد الكتابة

تُعرَّف المتحولة باسمها . وفي لغة الفورتران هذا الاسم هو عبارة عن سلسلة من 6 سمات أبجعددية على الأكثر (فقط من الأحرف والأرقام) . والسمة الأولى هي إلزامياً حرف .

أمثلة :

```
ALPHA
Z001
KILO etc...
```

1.32 ـ التحولات اللا توجهية (Variables Scalaires)

أ ـ التصريح الضمني

عملية إستعمال متحولة لمرّة واحدة على الأقل في أحد البرامج يكفي بأن يحجز لها المصرِّف عنواناً في الذاكرة .

في غياب أية دقة أخرى في التصريح ، ستُعتبر المتحولة في البرنامج كمتحولة .

ـ صحيحة

إذا كان الحرف الأول من إسمها هو : I, J, K, L, M, N .

ـ بفاصلة متحركة ودقة بسيطة

إذا كان الحرف الأول من الإسم يختلف عن I, J, K, L, M, N .

ب ـ التصريحات الجليّة (explicit declaration)

التصريحات الجلية (الواضحة) ، عندما تكون موجودة ، لها أفضلية عن القواعد السابقة . من الممكن إذاً التصريح عن متحولة كصحيحة أو بفاصلة متحركة ، أو بدقة مزدوجة ، أو مركبة ، أو منطقية .

التصريح عن المتحولة يجب أن يدخل إلى البرنامج قبل أية تعليمة تستعمل هذه المتحولة أو المتحولات المعتبرة . وبشكل عام ، نقوم بتجميع هذه التصريحات في بداية البرنامج .

متحولات صحيحة

التصريح : INTEGER liste

حيث liste هي عبارة عن لائحة بأسماء المتحولات ، مفصولة عن بعضها بواسطة فواصل . يمكن لهذه اللائحة أن تختزل في متحولة واحدة .

مثلاً :

```
INTEGER ALPHA, Z001
```

متحولات بفاصلة متحركة ودقة بسيطة

التصريح : REAL liste

متحولات بفاصلة متحركة ودقة مزدوجة

التصريح : DOUBLE PRECISION liste

متحولات مركبة

التصريح : COMPLEX liste

متحولات منطقية

التصرّيح : LOGICAL liste

1.33 ـ أبعاد المتحولات

من الممكن ، في لغة فورتران ، إستعمال جداول من 1 أو عدة دلائل (3 ثلاثة كحدّ أقصى للنظم الصغيرة ، وفي فورتران ANSI حتى 7 دلائل في بعض الحالات) .

تؤلف المتحولات جدولاً يحمل نفس الإسم ، وتراجع بواسطة هذا الإسم ، متبوعاً بلائحة من الدلائل بين أهلّة . وتكون جميعها من نفس النوع في جدول معين .

أ ـ التصريحات

وتتم ، قبل أي إلتقاء للمتحولة ، بواسطة الأمر DIMENSION :

التصريح : DIMENSION liste

مثلاً 1 : DIMENSION ALPHA (10,25)

ALPHA هي إذاً عبارة عن جدول من 250 = 25 × 10 متحولة حقيقية بدقة بسيطة . بلوغ ونيل كل متحولة يتم بواسطة إسمها إضافة إلى مؤشرين ، حيث المؤشر الأول (الدليل الأول) موجود بين الأعداد 1 و10 ، والثاني بين الأعداد 1 و25 .

مثل رقم 2 : DIMENSION X(100), K(1000)

X هي عبارة عن جدول من 100 متحولة حقيقية ، ببعد واحد (بمؤشر واحد)

K هي عبارة عن جدول من 1000 متحولة صحيحة ، وأيضاً ببعد واحد .

مثل 3 : DIMENSION TAB(30,30,5), M(10,10)

TAB هو عبارة عن جدول بثلاثة أبعاد ، ويتألف من 45 متحولة بالمجموع .

M هو عبارة عن جدول ببعدين ، ويتألف من 100 متحولة .

وعندما نرغب بتجاوز قواعد التصريح الضمني للنوع ، من الممكن أن نستعمل التصريحات مثل INTEGER ، REAL ، الخ . . . في هذه الحالة ، يجب أن يسبق التصريح عن النوع تصريحاً عن البعد .

مثلاً :

```
LOGICAL AIG
DIMENSION AIG(250)
```

بعض المصرِّفات تسمح بتكثيف التصريحين .

مثلاً :

```
LOGICAL AIG(250)
```

ب ـ الدليل (التأشير)

يُمكن أن يكون الدليل عبارة عن ثابتة صحيحة إيجابية ، أو متحولة صحيحة إيجابية ، أو عبارة جبرية من الشكل :

(متحولة + ثابتة) v + c

(ثابتة × متحولة)

v − c

c * v

c * v + ć

c * v − ć

بعض المصرفات يسمح بالتعابير العامة .

وفي الشك ، وللاحتفاظ بإمكانية نقل محتمل للبرنامج على أنظمة أخرى ، من الممكن أن ننحصر في إستعمال المتحولات الصحيحة المعرَّفة بشكل جيد .

وفي FORTRAN ، وعلى عكس ما يجري في أغلبية لغات البرمجة الأخرى ، وبالتحديد في COBOL و PL/ 1 ، فإن الجداول تُرتَّب في الذاكرة حسب مؤشرات (الدلائل indice) تدور من اليسار . هكذا ، مثلاً ، الجدول KT يُصرَّح عنه بالأمر .

```
DIMENSION KT(5, 3)
```

ورُتب على الشكل التالي :

KT(1, 1)   KT(2, 1)   KT(3, 1)   KT(4, 1)   KT(5, 1)   KT(1, 2)   KT(2, 2)
KT(3, 2)   KT(4, 2)   KT(5, 2)   KT(1, 3)   etc...

1.34 ـ إعداد المتحولات

من الممكن تصفير وإعداد المتحولات اللا إتجاهية (scalar) أو الجداول عند التصريف . نستعمل لهذا الغرض الأمر DATA .

الشكل العام للأمر DATA هو التالي :

Liste 1, List 2 ـ لوائح المتحولات

Liste de constantes ـ لائحة بالثوابت الممنوحة للمتحولات .

مثل 1 :

```
DATA X,Y,Z/0.1,25.7,1./
```

هكذا ، يتم إعداد المتحولة X إلى 0.1 وY إلى الثابتة 25.7 وZ إلى 1 .

مثل 2 :

```
DIMENSION K(100)
DATA K/50*0,25*2,25*1/
```

يهيىء ويُعدُّ الجدول الصحيح بالكامل : القيم الخمسون الأولى هي صفر ، وال 25 التالية هي 2 وأل 25 الأخيرة هي 1 . العدد 50 (مثلاً) متبوع بنجمة هو معامل للتكرار .

في حالة إعداد جدول من عدة دلائل ، يجب أن نتذكر نظام ترتيب مختلف عناصر الجدول في الذاكرة .

مثل 3 :

```
DIMENSION KT(5,3)
DATA KT/3*0,12*1/
```

هكذا ، فالعناصر KT (1,1) ، KT (2, 1) ، KT (3, 1) هي 0 والباقية تعادل 1 :.

مثل 4 :

```
LOGICAL C1,C2
DATA C1,C2/.TRUE.,.FALSE./
```

1.4 ـ أوامر اللغة

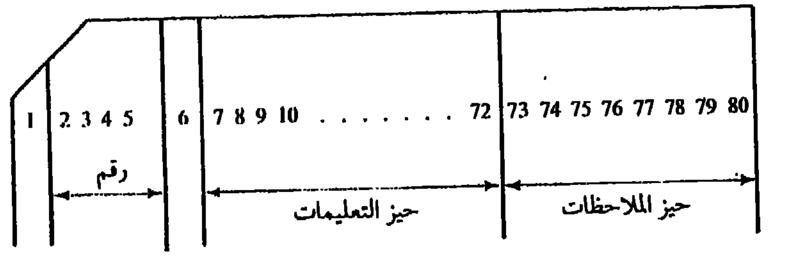
جميع أوامر اللغة هي عبارة عن كلمات أبجدية ، تناسب متتالية مُحدَّدة من العمليات . هكذا ، مثلاً ، GOTO ، STOP ، IF ، DO ، الخ . . .

وفي أغلب الأحيان ، يُشار إلى عمليات إنتقال حيز من الذاكرة من مكان إلى آخر ، إضافة إلى عمليات معالجة المعطيات بواسطة سمات خاصة .

أما كلمات لغة FORTRAN فهي غير محفوظة . ومن الممكن استعمالها كـأسماء متحولات . وهذا ما يجب ألا يُنصح به ، لأسباب تتعلّق بوضوح البرنامج .

2 ـ كتابة التعليمات

نكتب في لغة فورتران تعليمة واحدة على كل سطر . يتألف هذا السطر من عدة حيزات(*)



(*) العرض الجاري في موضوع البطاقة يبقى صالحاً في إستعمال الأداة الطرفية .

العامود 1 يُمكن أن يحتوي عند الاقتضاء على الحرف C . كامل النص الموجود على البطاقة يُمكن أن يعتبر ملاحظية وسيهمل بواسطة المصرِّف .

العامود من 5 -2 (ومحتملاً 5-1) خاص بكتابة عدد (رقم) صحيح ، إيجابي بدون إشارة ، يستخدم كوسم مرجعي للتعليمة . جميع الأرقام الموجودة في هذا الحيز يجب أن تكون مختلفة في نفس البرنامج .

العامود رقم 6 هو بشكل عام فارغ . وإذا كان يحتوي على سمة مختلفة موجودة في هذا العامود ، فالنص الموجود في الأعمدة 72-7 يعتبر تابعاً للسطر السابق .

الأعمدة 72-7 تُستعمل لكتابة نص التعليمة .

تهمل الأعمدة 80-73 بواسطة المصرِّف . ولكن ، بإمكاننا إذا أردنا ، إستعمالها لترقيم البطاقات .

هذه الحيزات يُمكن أن تكون مختلفة في حالة الأنظمة بالوقت المقسَّم ، وبالتحديد في كل ما يتعلَّق بعامود السلسلة . وفي أغلب الحالات نجد الحيِّز « رقم » ، الفارغ محتملاً ، متبوعاً بالحيز « تعليمة » .

3 ـ التركيبة العامة للبرنامج

البرنامج بلغة فورتران هو عبارة عن سلسلة من التعليمات ، المتبوعة بتصاريح التعليمة الأخيرة من البرنامج هي END . وفي الحقيقة هذه التعليمة هي موجهة نحو المصرِّف ، لتشير إليه وبواسطتها إلى النهاية الفيزيائية للبرنامج . وعند الاقتضاء ، يُمكن للبرامج الثانوية أن تتابع . وهي نفسها تنتهي أيضاً بأوامر من نوع END .

إدارة الذاكرة هي من نوع ساكن . وهذا يعني إن جميع الحيزات من الذاكرة والمناسبة للثوابت ، والمتحولات اللاتوجيهية أو غيرها ، إضافة إلى بعض تعليمات الوصف (مثلاً التعليمات FORMAT) هي محدَّدة بشكل ثابت قبل بداية تنفيذ البرنامج ولا تتعدَّل أبداً خلال هذا التنفيذ . لذا ، يجب على المبرمج في اللغة لغة FORTRAN أن يدرك هذا المفهوم في ذاكرته عند كتابة البرنامج . وبإمكاننا ، غالباً ، تفادي الاستعمال غير المفيد وإهدار مكان واسع من الذاكرة ، وذلك بإعادة إستعمال بعض العناصر من الجداول .

4 ـ تعليمات التبادل

يُرمز في لغة الفورتران إلى تعليمة التبادل بواسطة الرمز = أمثلة :

```
X = 12.5
K = J*L-1
```
الخ . .

يحسب التعبير لجهة يمين الاشارة = وتُنقل النتيجة إلى المتحولة على يسار الاشارة = . يُبدَّل المضمون السابق بواسطة المضمون الجديد . مهما يكن نوع التعبير إلى يمين الإشارة=، تحـوَّل القيمة الناتجة ، ومحتملاً ، إلى نوع يتناسب مع نوع المتحولة الهدف ، الى اليسار .

أمثلة :

```
X = 12.337
M = X
X = M
```

التعليمة (1) تنقل القيمـة 12.337 ، الحقيقية الإيجـابية ، إلى المتحـولة X الحقيقية أيضاً (نفترض عدم وجود تصريح واضح عن X ، ولا أيضاً عن M) . هذه التعليمة (1) لا تؤدي الى أي عملية تحويل .

التعليمة (2) تؤدي ، مع تحويل من الحقيقي إلى الصحيح ،إلى تبديل قيمة X ، وهي 12.337 ، بقيمة M . فقط القسم الصحيح سيوجد في M ، الـذي يأخـذ القيمة الصحيحة 12 .

التعليمة (3) تنقل ، مع تحويل العدد الصحيح إلى حقيقي ، قيمة المتحولة M ، أي 12 ، إلى X ، التي تأخذ الآن القيمة الحقيقية 12 .

5 ـ العمليات الجبرية

5.1 ـ المؤثرات الجبرية

المؤثرات الجبرية (arithmetic operations) هي التالية :

+ الجمع

− الطرح

* الضرب

/ القسمة

** مرفوع ب .

أمثلة :

مجموع X مع Y	X + Y
الفرق K ناقص 1 .	K - 1
نتيجة ضرب ALPHA مع 3.5	ALPHA * 3.5
قسمة GAMMA على 5.1	GAMMA/5.1
X1 بقوة K .	X1 ** K

وفي الأصل ، التعبير الجبري ، أي النلسلة من المتحولات والثـوابت المفصولـة

بواسطة مؤثرات ، يجب أن يكون متجانساً : المتحولات والثوابت يجب أن تكون بنفس النوعية ، صحيحة ، أو بفاصلة متحرِّكة ، أو مركبة ، الخ . . .

وبسبب ، كون أغلبية المصرِّفات تقبل خلط هذه الأنواع . في هذه الحالة ، فإن حسابة قيمة التعبير تتم في نوع المتحولة أو الثابتة التي تُقدِّم الدقة الأكبر . أي ما معناه .

في دقة مزدوجة أو مركبة
أو بالغلط في الحقيقي بدقة بسيطة .
أو بالغلط في الصحيح .

5.2 ـ الدالات

هنـاك نوعان من الدالات في لغـة فورتـران : الدالات الـرياضيـة التي تناسب خوارزميات الحساب المعقدة ، ودالات الخدمة ، المناسبة لمعالجة المعطيات البسيطة .

يُمكن للدالة أن تكون بواحد أو عدة مُعاملات ، موضوعة بين أهلـة . المُعامل هو تعبير جبري ، كل دالة تعيد قيمة واحدة .

5.21 ـ الدالات الجبرية

القواعد

أ ـ لا يُمكن أن تستعمل إسم دالة رياضية كإسم لمتحولة .

ب ـ الدالة الرياضية لا تستعمل أبداً النوع الصحيح ، أو النوع المنطقي . الأنواع الوحيدة المستعملة هي الحقيقية ، بدقة بسيطة أو مزدوجة ، والنوع المركب .

ج ـ في الحالة التي نستعمل بها دالة رياضية بدقة مزدوجـة أو مركبـة ، يجب من جهة ، إستعمال دالة تحت الاسم المناسب لنوعها ، ومن جهة أخرى التصريح عنها بوضوح في أمر من نوع DOUBLE PRECISION أو COMPLEX .

د ـ النتيجة التي تعيدها الدالة هي من نفس نوع مُعاملاتها .

أمثلة :

جذر تربيعي من X	SQRT
دالة أسية من Z- 1.5	EXP(Z-1.5)
جيب A * .15	SIN(15.*A)

الجدول 1 يعطينا لائحة بالدالات الرياضية الرئيسية الموضوعة بتصرُّف فورتران . الدالات التي يبدأ إسمها بالحرف C هي مركبة (ما عدا COS, COSH) . وتلك التي تبدأ أسمائها بـ D هي بدقة مزدوجة .

جدول 1 ـ الدوال الرياضية الرئيسية

النوع	المعنى	الدالة
حقيقي	جذر تربيعي من X	SQRT (X)
دقة مزدوجة	جذر تربيعي من X	DSQRT (X)
مُركّبة	جذر تربيعي من X	CSQRT (X)
حقيقي	لوغاريتم نابير من X	ALOG (X)
دقة مزدوجة	لوغاريتم نابير من X	DLOG (X)
مركب	لوغاريثم نابير من X	CLOG (X)
حقيقي	لوغاريتم عشري من X	LOG 10 (X)
حقيقي	دالة أسية e^X	EXP (X)
حقيقي	جيب X (Radians)	SIN (X)
حقيقي	جيب X (درجات)	SIND (X)
حقيقي	جيب التمام X (درجات)	COS (X)
حقيقي	تضل X	ATAN (X)
حقيقي	جيب زائدي المقطع من X	SINH (X)
حقيقي	جيب التمام زائدي المقطع من X	COSH (x)
حقيقي	مولّد الأعداد العشوائية	RAND (X)
حقيقي	دالة خطأ	ERF (X)
حقيقي	جذر مُكعب	CBRT (X)

ملاحظة : لائحة الدالات الرياضية الجاهزة تتعلّق كثيراً بالمصرِّف المستعمل . ويجب مراجعة الوثائق المناسبة .

5.22 ـ دالات الخدمة

القاعدة :

يجب تفادي التصريح عن إسم دالة خدمة ، لا ضمنياً ولا بوضوح ، هكذا غلطة ستؤدي إلى أن يصبح هذا الإسم إسماً لمتحولة وسيكون من المستحيل إستعمال الدالة المناسبة في البرنامج .

الجدولُ 2 يعطي لائحة بالدوال الرئيسية للخدمة والجاهزة بتصرُّف فورتران .

جدول رقم 2 ـ الدوال الرئيسية للخدمة

نوع نتيجة	معامل	معنى	دالة
حقيقي	حقيقي	قيمة مطلقة	ABS (X)
صحيح	صحيح	قيمة مطلقة	IABS (X)
حقيقي	صحيح	تحويل من صحيح إلى حقيقي	FLOAT (X)
صحيح	حقيقي	تحويل من حقيقي إلى صحيح	IFIX (X)
حقيقي	مركب	قسم حقيقي من عدد مركب	REAL (X)
حقيقي	مركب	قسم وهمي من عدد مركب	AIMAG (X)
صحيح	حقيقي	قسم صحيح من عدد حقيقي	INT (X)
حقيقي	حقيقي	قسم صحيح من حقيقي	AINT (X)
حقيقي	حقيقي	يعيد الإشارة (1 + ، 1 - أو 0) .	SIGN (X)
حقيقي	صحيح	يعيد المعامل الأكبر	AMAX0 (X, Y, ...)
حقيقي	حقيقي	يعيد المعامل الأكبر	AMAX1 (X, Y, ...)
صحيح	صحيح	يعيد المعامل الأكبر	MAX0 (X, Y, ..)
صحيح	حقيقي	يعيد المعامل الأكبر	MAX1 (X, Y, ..)
حقيقي	صحيح	يعيد المعامل الأصغر	AMIN0 (X, Y, ...)
حقيقي	صحيح	الباقي من قسمة X على Y	MOD (X, Y)
حقيقي	صحيح	الباقي من قسمة X على Y	AMOD (X, Y)

ملاحظة : اللائحة المعروضة هنا تتعلّق بنوعية المصرِّف المستعمل لذا يُرجى مراجعتها في الوثائق التابعة له .

5.3 ـ قواعد الأفضلية

5.31 ـ أولوية مؤثرات الدالات

في تعبير لا يحتوي على أهلة ، ما عدا محتملاً تلك التي تتبع إسم الدالة ، تتم الحسابات من اليسار نحو اليمين ، مع المحافظة على الأولويات التالية :

(1) طلب الدالة .

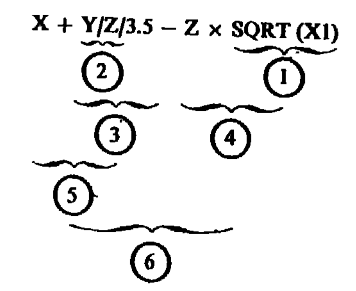
(2) أسية (exponentiel)

(3) الضرب والقسمة .

(4) الجمع والطرح .

مثلاً :

```
X + Y/Z/3.5 - Z * SQRT (X1)
```



تُنفَّـذ الدالة SQRT أولاً . بعد ذلك تحسب القسمة الأولى (Y / Z) . هذه النتيجة الجزئية تقسم على 3.5 وتحتفظ بنتيجة جزئية رقم 3 . بعد ذلك نقوم بإجراء عملية ضرب Z* بنتيجة حساب SQRT . يبقى حساب المجموع الأول (5) وطرح النتيجة الجزئية (4) في النتيجة (5) .

5.32 ـ إستعمال الأهلّـة

يُمكن تعديل قواعد الأولوية باستعمال الأهلة .

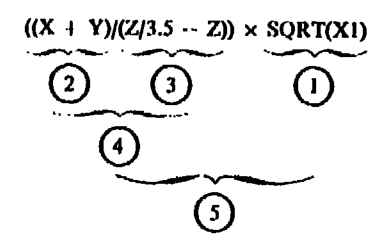
القواعد :

ـ في تعبير جبري ، عدد الأهلة المفتوحة يعادل عدد الأهلة المُغلقة .

ب ـ الأهلة الأكثر داخلية هي التي تُنفَّـذ أولاً .

مثلاً :

```
((X + Y)/(Z/3.5 - Z)) * SQRT(X1)
```



وكما في المثل السابق ، تُنفّـذ الدالة أولاً . بعد ذلك تحسب على التوالي التعابير (Y + X) ، وبعد ذلك (Z – 3.5 / Z) . في الحالة الأخيرة ، تحسب القيمة أوّلاً ونطرح 2 من نتيجة القسمة المحسوبة . وبعد ذلك نقسم النتيجة الجزئية رقم 2 على النتيجة الجزئية رقم 3 . نحصل على النتيجة رقم 4 ، الذي نضربه بنتيجة الحساب SQRT .

6 ـ العمليات المنطقية

هنـاك طبقتان من العمليـات المنطقيـة : عمليات المقـارنة والعمليـات المـنطقيـة الأساسية .

وكالثوابت والمتحولات المنطقية ، فالتعابير المنطقية لا يُمكن أن تأخذ سوى قيمتين « حقيقة TRUE » و« غلط FALSE » .

6.1 ـ مؤثرات المقارنة

الجدول رقم 3 يُوجز عمليات المهام المنطقية للمقارنة .

ملاحظة : جميع المؤثرات المنطقية هي في لغة فورتران مسبوقة بواسطة نقطة .

الجدول 3 ـ مؤثرات منطقية للمقارنة

قيمة	مثلاً	تسمية	مؤثر
حقيقة إذا A = B	A.EQ.B	يعادل	.EQ.
حقيقة إذا	A i B .NE.	A.NE.B	يختلف عن
حقيقة إذا A > B	A.GT.B	أعلى من	.GT.
حقيقة إذا A ≥ B	A.GE.B	أعلى أو يعادل	.GE.
حقيقة إذا A < B	A.LT.B	أقل من	.LT.
حقيقة إذا A ≤ B	A.LE.B	أقل أو يعادل	.LE.

تجرى العمليات المنطقية للمقارنة على متحولات من نفس النوع . جميع الأنواع هي مسموحة ، ما عدا المتحولات المنطقية والمركبة . وبشكل خـاص ، من الجائـز مقارنـة متحولتين من نوع سلسلة من السمات : المقارنة ستتم بمراجعة التعريف الثنائي الداخلي للسمات ، حسب النظام المستعمل (أنظر الملحق A ، الجداول ASCII وEBCDIC)*.

(*) ملاحظة : أغلب المصرفات تسمح باستعمال.EQ. و.NE. للمتحولات المنطقية والمركبة .

6.2 ـ مؤثرات منطقية أساسية

وهي تناسب المهام الإلكترونية الأساسية لدارات الحاسبات .
الجدول 4 يختصر هذه الدالات .

6.3 ـ التعابير المنطقية

6.31 ـ أولوية المؤثرات المنطقية

ترتيب الأولوية هو التالي :
(1) مؤثرات منطقية للمقارنة .
(2) .NOT .
(3) .AND .
(4) .OR .

الجدول 4 ـ مؤثرات منطقية أساسية

قيمة	مثلاً	تسمية	مؤثر
حقيقة إذا كان L1 غلط	.NOT. L1	إعكاس	.NOT.
حقيقة إذا كان كلا L1 وL2 حقيقة . غلط في جميع الحالات الأخرى	L1. AND. L2	تقاطع (و)	.AND.
حقيقة إذا كان L1 أو L2 حقيقة . وغلط إذا كان كلاهما غلط	L1. OR. L2	أو (جمع)	.OR.

6.32 ـ التعابير المنطقية

التعابير المنطقية هي شبيهة بالتعابير الجبرية . وحسابها يتم من اليسار نحو اليمين ، مع اعتبار الأولوية . وكما هي الحال بالنسبة للتعابير المنطقية ، فإن الأولوية يمكن أن تكون مفروضة باستعمال الأهلّة . إستعمال الأهلة يخضع إذاً لنفس القواعد كما في التعابير الجبرية .

مثلاً :

```
LOGICAL L1,L2
L1 = A.EQ.B .AND.L2 .OR. X.EQ.Y
```

L1 هي حقيقة إذا كانت ، في نفس الوقت ، المتحولة A تعادل B وL2 حقيقة ، من جهة ، أو من جهة أخرى إذا كانت X تعادل Y ، أو أيضاً إذا كانت التعابير الثلاثة A.EQ.B ، L2 وX.EQ.Y هي حقيقة .

7 ـ التحكّم بدوران وتنفيذ البرنامج .

تسمح أوامر التحكّم بدوران البرنامج بقطع ، حسب حاجات الخوارزم ، الدوران التسلسلي الطبيعي للبرنامج .

وتتألف من نوعين : أوامر الدوران غير المشروطة وأوامر الدوران المشروطة .

7.1 ـ أوامر الدوران غير المشروطة

أ ـ GOTO البسيطة

الشكل العام :

```
GO TO étiquette
```

مثلاً :

```
GO TO 253
```

القواعد :

الوسم etiquette يجب أن يكون إلزامياً موجوداً لمرّة واحدة فقط في البرنامج ، وفي الحيّز المحفوظ لهذه الغاية (الأعمدة من 1 إلى 5) .

ب ـ الأمر GOTO المحسوب

الشكل العام :

```
GO TO (e1, e2, ..., eN), v
```

e1, e2, ..., eN هي عبارة عن أوسمة ، ليس من الضروري أن تكون مختلفة . v هي عبارة عن متحولة صحيحة ، لا يمكن أن تأخذ سوى قيم إيجابية موجودة بين 1 وN .

مثلاً :

```
GO TO(1,253,27,1,12), K
```

القاعدة :

عند تنفيذ GOTO يجب إلزامياً أن تأخذ K قيمة صحيحة موجودة بين 1 و5 ، في هذا المثل .

ج ـ الأمر GOTO الممنوح .

الشكل العام :

```
GO TO v, (e1, e2, ..., eN)
```

e1, e2, ..., eN هي عبارة عن أوسمة ، ليست بالضرورة مختلفة . v هي عبارة عن متحولة صحيحة مُعدّة إلزامياً بواسطة أمر ASSIGN .

مثلاً :

```
GO TO Y, (12,27,2344,1,8,815)
```

(أنظر الفقرة التالية لكيفية تشغيل هذا الأمر) .

د ـ الأمر ASSIGN

الشكل العام :

ASSIGN étiq TO v

v هي عبارة عن متحولة صحيحة للأمر GO TO الممنوح .

etiq هي عبارة عن وسم من لائحة الأمر GO TO الممنوح .

مثلاً :

```
ASSIGN 2344 TO Y
```

بعد تنفيذ هذا الأمر ، يصبح تنفيذ الأمر السابق GO TO الممنوح يتم كما في الأمر السابق .

ملاحظة : لا يجب أن نخلط بين GO TO المحسوب و GO TO الممنوح . في الحالة الأخيرة ، لا تحتوي المتحولة v على القيمة الجبرية 2344 (مثلاً) ولكن على عنوان من الذاكرة . في الحالة الأولى ، تحتوي المتحولة على قيمة جبرية عادية .

قاعدة مشتركة بين جميع أوامر التفريع(*) . .

يجب على أمر التفريع غير المشروط أن يكون دائماً متبوعاً بتعليمة مُزوَّدة بوسم . وإذا كانت التعليمة التالية للأمر GO TO بدون وسم ، فلن يُوجد أية وسيلة للوصول إليها .

7.12 ـ وقف البرنامج

أ ـ STOP

هذا الأمر يُوقف البرنامج بشكل نهائي ويعيد التحكُّم إلى المشرف . ويُمكن أن يوضع في أي مكان من البرنامج .

ب - PAUSE

في بعض الأنظمة ، يُوقف هذا الأمر فعلياً الحاسب ويطبع رسالة متبوعة برقم ، على لوحة التحكّـم الخاصة بالآلة .

نكتب إذاً PAUSE numero ، حيث numero هي عبارة عن ثابتة صحيحة إيجابية مُؤلّـفة من 5 أرقام على الأكثر . من الممكن إذاً إجراء عدة عمليات : تركيب الشـريط المغناطيسي ، تغيير ورق الطابعة ، الخ . . .

في حـالة الأنـظمة بـالبرمجـة المتعددة ، يضـع الأمر PAUSE القسم المعتمـد في الانتظار ، والمكنة تعمل داخلياً بشكل طبيعي .

(*) مهما تكن لغة البرمجة المستعملة .

ج - END

هذا الأمر هو فيزيائياً الأخير في البرنامج . وهو ينهي أيضاً كل برنامج - ثانوي .
وفي الحقيقة ، يقوم في الغالب بالإشارة إلى المصرِّف بأن أية تعليمة فورتران لا تتبع : والمصرِّف يُمكنه إذاً أن يشرع بمراحل ترجمة البرنامج وترتيب البرنامج الموضوعي القابل للتنفيذ .

7.2 ـ أوامر الدوران المشروطة

7.2.1 ـ الأمر IF المنطقي

الشكل العام هو :

IF (expression logique) instruction

الأهلّة هي إلزامية . التعليمة instruction هي عبارة عن تعليمة قابلة للتنفيذ ، ما عدا التعليمة DO أو التعليمة IF المنطقية . ولا يُمكن أن تكون تعليمة تصريح .

expression logique عبارة عن تعبير جبري أو معادلة جبرية .

مثلاً :

```
IF(A.NE.B) GO TO 253
```

قاعدة :

ـ إذا كان التعبير المنطقي حقيقة ، سيتم تنفيذ التعليمة المذكورة في الأمر بواسطة الوسم والبرنامج سيُتابع بشكل طبيعي بعد ذلك .

ـ إذا كان التعبير المنطقي غلط ، فستهمل التعليمة والبرنامج سيتابع تنفيذه على التوالي .

7.2.2 ـ الأمر IF الجبري

الشكل العام :

IF (expression arithmetique) e1, e2, e3

الأهلة هي إلزامية . والتعبير الجبري (expression arithmetique) يُمكن أن يكون عبارة عن متحولة بسيطة .

القاعدة :

ـ إذا كان التعبير الجبري هو عبارة عن قيمة سلبية ، فهناك تفريع الى الوسم e1 .

ـ إذا كان التعبير الجبري يعادل صفر ، فسيتم تفريع إلى الوسم e2 .

ـ إذا كان التعبير الجبري إيجابياً ، فهناك تفريع إلى الوسم e3 .

مثلاً :

```
IF(A+B/C) 12,44,253
```

8 ـ حلقات البرنامج

8.1 ـ الشكل العام

تستخدم الحلقة من البرنامج لتكرار سلسلة من التعليمات لعدة مرّات . لهذه الغاية يُستعمل الأمر DO .

الشكل العام هو :

```
DO  étiq v = p1, p2, p3
      . Séquence
      . d'instructions
étiq CONTINUE
```

étiq ـ عبارة عن وسم معين .

Séquence d'instruction : سلسلة من التعليمات .

étiq هي عبارة عن وسم من البرنامج ، واحد ، يُوضع أمام تعليمة مُحايدة هي CONTINUE أو أمام تعليمة مختلفة عن تعليمة التفريع المشروط ، أو STOP ، أو أي تعليمة DO أخرى .

v عبارة عن متحولة صحيحة ، تدعى دليل الحلقة .

p1, p2, p3 تمثّل المتغيّرات الوسيطة للحلقة : وهي عبارة عن ثوابت صحيحة بدون دليل ، بقيم صحيحة إيجابية . المُتغيّر p3 هو إختياري . وإذا أُهمل ، فهو يأخذ ضمنياً القيمة 1 : وهو عبارة عن خطوة الحلقة .

التشغيل : سيتم تنفيذ سلسلة التعليمات لمرّات عديدة ، وللقيم v من p1 إلى p2 وبخطوة تعادل p3 (أو بخطوة تعادل 1 إذا كانت p3 غير مُسجّلة) .

مثل 1 :

```
DO 15 K = 1,50
   .
   .
   .
15 CONTINUE
```

الحلقة ستدور 50 مرّة ، ولـ K تعادل 50, ..., 1,2 .

مثل 2 :

```
DO 12 J = 2,19,3
   .
   .
12 CONTINUE
```

الحلقة ستدور لجميع القيم التالية لـ J : 2, 5, 8, 11, 14, 17

8.2 ـ القواعد

1 ـ إختبار نهاية الحلقة يتم في مستوى الأمر CONTINUE . تدور الحلقة على الأقل مرة واحدة .

2 ـ لا يُمكن الدخول الى حلقة إلا في مستوى الأمر DO .

3 ـ من الممكن الخروج من الحلقة (بواسطة GO TO أو IF) في أي مكان منها .

4 ـ يمنع تعديل قيمة الدليل بداخل الحلقة .

5 ـ من الممكن تداخل عدة حلقات فيما بينها . ومن الممكن إستعمال نهاية حلقة مشتركة بين عدة حلقات .

6 ـ من الممكن الخروج من حلقة وبعد ذلك الدخول الى الحلقة وفي التعليمة التي تتبع مباشرة تعليمة الخروج (هذه الطريقة في العمل لا يُنصح بها : فهي تؤدي إلى عدم وضوح في البرنامج) .

7 ـ من الممكن دعوة برنامج ثانوي من داخل الحلقة .

8 ـ لا يُمكننا أبداً إعادة تحديد مُتغيِّرات الحلقة خلال تنفيذه .

مثلاً : تصفير جدول ببعدين

```
        DIMENSION TAB(15,30)
        .
        .
        DO 100 K = 1,15
        DO 100 J = 1,30
        TAB(K,J) = 0.
   100- CONTINUE
```

لقد إستعملنا هنا حلقتين متداخلتين . بإمكاننا هنا عدم إستعمال CONTINUE ووضع الوسم 100 ، كنهاية مشتركة لحلقتين أمام التعليمة TAB (K, J) = 0 .

9 ـ إدخال ـ إخراج

9.1 ـ الشكل العام

الشكل العام لأمر بالإدخال ـ الإخراج هو التالي :

Ordre(*u*, *f*, opt1, opt2) liste

ـ ordre يدل على إتجاه العملية : الادخال (القراءة) أو الاخراج (الكتابة) . الأمر الموجود هنا يُمكن أن يكون READ للقراءة ، أو WRITE للكتابة .

ـ u هو الرقم المنطقي للوحدة الفيزيائية المُعتمدة : قارىء البطاقات ، لوحة ملامس الأداة الطرفية ، طابعة جهاز الأشرطة المغناطيسية ، الخ . u يعادل ثابتة صحيحة إيجابية بدون إشارة ، أو متحولة صحيحة بقيمة إيجابية .

قيمة u تتعلّق بالنظام المستعمل

ـ f هي عبارة عن وسم التعليمة FORMAT .

ـ opt 1 ، إختياري ، وعلى شكل ERR = n1 ، حيث n1 هي وسم التعليق في حالة حدوث خطأ في القراءة .

ـ opt2 ، إختياري ، وعلى الشكل END = n2 ، حيث n2 هي عبارة عن وسم التعليق في حالة قراءة نهاية سجل الإدخال .

(opt1, opt2 ليست بتصرُّف جميع الأنظمة ، ولا تستعمل إلا مع الأمر READ ، وهي إختيارية ويُمكن أن يُبدَّل ترتيبها) .

ـ liste هي عبارة عن لائحة بالمتحولات ، وتكون عادةٍ إختيارية .

أمثلة :

```
READ(5,150) X,Y,Z,K
WRITE(6,212) ALPHA,BRAVO
```

9.2 ـ حالة خاصة للادخال والاخراج الكلاسيكي

9.2.1 ـ القراءة

الأمر العام للقراءة هو :

READ(*u*, *f*) liste

وأغلب الأنظمة تسمح باستعمال الكتابة المعادلة :

READ *f*, liste

مع نفس الاتفاقات بالنسبة لـ f ، وسم نسق ، واللائحة .

9.2.2 ـ الكتابة

الأمر العام للكتابة هو :

WRITE (*u*, *f*) liste

بنفس الطريقة ، فإن أغلب الأنظمة تسمح باستعمال .

PRINT *f*, liste

في بعض الحالات ، من الممكن طلب تثقيب البطاقات .

وفي أغلب الأحيان من الممكن إستعمال الأمر المُعادل :

PUNCH *f*, liste.

9:3 ـ حالة خاصة بالنسبة للنواقل المغناطيسية

الأوامر العامة تستعمل أيضاً للادخال والاخراج من النواقل المغناطيسية ، أشرطة وإسطوانات . وهناك أيضاً ، بنفس الطريقة كما في البطاقات والطابعات ، يوجد تحويل من

شكل خارجي إلى شكل داخلي والعكس بالعكس .

من الممكن أن نربح وقتاً بإلغاء هذا التحويل المزدوج : نستعمل إذاً الأوامر بدون نسق :

ordre(*u*) liste

حيث u تمتاز بنفس المعنى كما في الحالة العامة .

أمثلة :

```
WRITE(1) Y,Y,TAB
READ(2) ALPHA,M1,M2
```

9.4 ـ أوامر التحكّم بالنواقل المغناطيسية .

أ ـ REWIND u

هذا الأمر يناسب إعادة لف الوحدة المنطقية u . ويناسب فتح السجل المناسب . ومن الممكن إستعمال العدد الذي نرغب به من المرّات في نفس البرنامج ، إذا رغبنا بإعادة إستعمال نفس السجل ، كما في القراءة كذلك في الكتابة .

ب ـ ENDFILE U

هذا الأمر يستعمل لوضع علامة نهاية التسجيلة على السجل الذي سنقوم بكتابته على الوحدة المنطقية u .

ج ـ BACKSPACE u

هذا الأمر يؤدي إلى تحريك خلفي للوحدة المنطقية u إلى طول يعادل طول التسجيلة الأخيرة المقروءة أو المكتوبة .

9.5 ـ اللائحة

9.5.1 ـ تعريف

اللائحة هي عبارة عن سلسلة من المتحولات المؤشرة أو الغير مؤشرة ، من جميع الأنواع ، مفصولة بواسطة فواصل .

في الكتابة ، تعالج المتحولات الواحدة بعد الأخرى ، من اليسار إلى اليمين حتى الانتهاء من اللائحة .

9.5.2 ـ الجداول

من الممكن أن يدخل واحد أو عدة جداول في اللائحة . بإمكاننا إذاً ترقيم العناصر بواسطة حلقات ضمنية .

مثلاً :

```
READ(5,150) (TAB(K), K = 1,15)
READ(5,150) TAB(1), TAB(2),...,
                      TAB(15)
```

الأشكال (1) و(2) من هذا المثل هي متعادلة بالكامل . الشكل (1) يستعمل حلقة ضمنية . في هذه الحالة الخاصة ، بإمكاننا أيضاً أن نكتب :

```
READ(5,150) TAB
```

إذا كان الجدول TAB مصرَّحاً عنه أيضاً .

```
DIMENSION TAB(15).
```

القواعد :

في حالة إستعمال الحلقات الضمنية ، فإستعمال الأهلة الخارجية هو إلزامي .

مثلاً :

```
DIMENSION M(10,10)
.
.
READ(5,211) ((M(K,J),J = 1,10),
                     K = 1,10)
```

في الحالة التي نستعمل الجدول باسمه في اللائحة فقط ، بدون دليل ، فيجب أن نتذكر قواعد تسجيل الجداول في الذاكرة بلغة فورتران (فقرة 1.33, b) .

هكذا ، فالكتابة التالية :

```
WRITE(6,313) M
```

هي معادلة لـ :

```
WRITE(6,313) ((M(K,J),K = 1,10),
                    J = 1,10)
```

الحلقة الضمنية الداخلية هي متداخلة في الأخرى .

9.6 النسق Format

9.6.1 ـ تعريف الشكل العام

مهمة النسق هي وصف الشكل الخارجي للمعطيات المطلوب قراءتها أو كتابتها :

والشكل العام هو التالي : (étiq FORMAT (Liste de spécification

القواعد :

étiq هي عبارة عن وسم إجباري .

Liste de spécification هـو عبارة عن لائحـة مواصفـات النسق (الشكـل) ، مفصولة بواسطة فواصل .

مواصفات النسق (الشكل) هي نوعين :

أ ـ التصفيح mise en page

توضيح	مواصفة
العبور إلى التسجيلة التالية	/
إنشاء سلسلة سمات بطول n	nH
طفور (ترك) عن n سمة أو إدخال n سمة بياض	nX
مثل nH ولكن بين أبوستروف	'.....'

التكرار : مواصفات التصفيح يُمكن أن تكون مسبوقة بمُعامِل تكرار ، ثابتة صحيحة بدون إشارة . يجب أن تكون بين هلالين .

مثلاً :

```
5(/)
12(2H+*) etc...
```

ب ـ معالجة المعطيات

توضيح	مواصفات
حيِّز معطى صحيح ، بطول n سمة	In
حيِّز معطى حقيقي (عدد حقيقي)، بطول n سمة ، حيث m منها للجزء الكسري	Fn.m
حيِّز معطى حقيقي بفاصلة متحركة ، بطول n سمة، حيث m للجزء الكسري	En.m
حيِّز معطى حقيقي ، بدقة مزدوجة بطول n سمة، حيث m منها للجزء الكسري	Dn.m
حيِّز معطى من سلسلة من السمات بطول n	An
حيز معطى منطقي ، بطول n	Ln
حيز معطى بالنظام الثماني ، وبطول n	On

تكرار : مواصفات معالجة المعطيات يمكن أن تكون مسبوقة بمعامل للتكرار ، أو

ثابتة صحيحة بدون إشارة الأهلة ـ ليست إلزامية .

مثلاً :

```
5  F  6.2
10  I  15 etc...
```

9.6.2 ـ مواصفات التصفيح

أ ـ المواصفة /

وهي تتحكّم بالطفور عن تسجيلة

عند الإدخال ، نقفز عن بطاقة ، أو ، بشكل عام عن تسجيلة .

عند الإخراج ، نقفز عن سطر ، أو بشكل عام عن تسجيلة .

ب ـ المواصفة X

وتستعمل بالشكل nX ، حيث n هي ثابتة صحيحة بدون إشارة (ولا تكون أبداً متحولة) .

عند الإدخال ، تهمل المكنة n سمة ، أو n عامود من البطاقة .

عند الإخراج ، يجري إدخال n بياض .

ج ـ مواصفات إدخال سلاسل السمات

nH . . . تسمح ، عند الإخراج ، بإدخال أو إدراج سلسلة من السمات بطول n .

من الأسهل إستعمال ، إذا كان هذا ممكناً ، زوج من الفواصل العليا (' ') ، كما يجري عند وصف ثابتة من نوع سلسلة من السمات .

هذه المواصفات ليس لها أي معنى عند الإدخال ، ولكنها ليست غلط . وبإمكانها أن تحجز حيِّزاً من نوع سلسلة من السمات دون أن يكون هناك حاجة لتعريف متحولة خاصة : من الممكن إذاً إعادة تعريف الحيِّز .

9.6.3 ـ مواصفات معالجة المعطيات

أ ـ المواصفات I (أعداد صحيحة)

الشكل هو : In ‎(Integer = I)‎ .

n هي عدد السمات المستعملة ، مع تعداد الإشارة الحسابية المحتملة .

عند الإدخال ، يجب تركيز العدد لجهة اليمين . الإشارة الحسابية ، إذا كانت موجودة (الاشارة ناقص (−) هي إلزامية للأعداد السلبية) ، يجب أن تسبق مباشرة الرقم الأكبر دلالةً في العدد . وفي جميع الحالات ، وهذه قاعدة عامة لجميع المواصفات ، تؤخذ في الاعتبار ال n سمة فقط . جميع السمات الأخرى كالبياض ، الإشارة + أو − أو

الأرقام تؤدي إلى إثارة خطأ في القراءة . عند الإخراج ، يُركّـز العدد لجهة اليمين . لا تطبع الاشارة إلا إذا كان العدد سلبياً .

ب ـ المواصفة F (أعداد حقيقية)

الشكل هو : (Float = (F) Fn. m

n هو العدد الكامل للسمات المستعملة ، مع تعداد الإشـارة المحتملة والفاصلة العشرية ، إلزامياً .

m هو عدد الأرقام بعد الفاصلة العشرية .

عند الإدخال ، التسطيريتم على الفاصلة العشرية . كل سمة أخرى مثل البياض ، الإشارة + أو − ، الفاصلة العشرية الوحيدة والأرقام تؤدي إلى ظهور خطأ في القراءة .

عند الإخراج ، يُسطّـر العدد على الفاصلة العشرية (يُركّـز العدد) .

لا تُطبع الإشارة إلا إذا كان العدد سلبياً .

ج ـ المواصفة E (أعداد حقيقية)

الشكل : En.m (E مثل أس) .

n هي العدد الكامل للسمات المستعملة ، مـع تعداد الإشـارتين (إشـارة الجزء العشري (mantisse) ، وإشارة الأس (Exposant)) ، الفاصلة العشرية للجزء العشري هي إلزامية ، والحرف E يعني « قوة عشرة أو أس) .

m هي عدد الأرقام بعد الفاصلة العشرية في الجزء العشري . وهذا الأخير هو بين 0 و1

عند الإدخال ، يسمح بحرية أكبر لتمثيـل الأعداد . وفي أغلب الأحيـان ينصح بكتابة الأعداد في الشكل المعيَّـر (normalisid) :

مثلاً :

```
- 0.123456E-03
```

الإشارة لا تكتب إلا إذا كان العدد سلبياً ، والجزء العشري هو بين 0.1 و1.0 في القيمة المطلقة . ولدقة أكبر تتعلق بالإمكانيات المقدمة ، يرجع القارىء إلى المخطوطات المناسبة للمصرِّف الذي يستعمله . عند الإخراج ، التمثيل يكون عادة بشكل مُعيَّـر .

قاعدة :

يجب أن نستعمل المواصفة E عند الإخراج في كل مرّة نرغب فيها بطباعة الأعداد الحقيقية حيث نهمل مقدار القيمة .

د ـ المواصفة D (أعداد حقيقية)

تُستعمل هذه المواصفة في الشكل الذي تُستعمل فيه المواصفة E ، ولكنها تتعلّـق

بالأعداد بدقة مزدوجة . وهي تتبع نفس القواعد ، مع تقريباً تبديل الحرف E بالحرف D ، وذلك في الإخراج والادخال .

هـ ـ المواصفة A (سلاسل السمات)

الشكل An (A من Alphanumérique) .

n هي العدد الكامل للسمات المستعملة

عند الإدخال ، n تساوي القيمة القصوى المناسبة للعدد الأقصى من السمات القابلة للتخزين في كلمة من الذاكرة ، مثلاً : لمكنة بكلمة من 32 بتة .

طبعاً ، يُمكن لـ n أن تأخذ قيماً أصغر .

عند الإخراج ، يُمكن أن تكون n مختلفة . يُكمل الحيِّز بواسطة فراغات لجهة اليمين .

و ـ المواصفة L (معطيات منطقية)

الشكل هو : Ln (L من Logique) .

n تعادل بشكل عام 1 .

عند الإدخال ، السمات الوحيدة المعروفة هي T وF ، « حقيقة TRUE » و« غلط FALSE » . وفي أغلب الأحيان يجري التصرُّف على البياض وكأنه « غلط T » .

عند الإخراج ، السمات المطبوعة هي فقط T وF .

ز ـ المواصفة 0 (جميع المعطيات)

الشكل هو : On (O من Octal) .

n هي الطول الكامل للسمات ، في الحيِّز المستعمل .

عند الإدخال ، السمات الوحيدة المسموح بها هي الأرقام من 0 إلى 7 إضافة إلى البياض التي تعالج كصفر (0) .

عند الإخراج ، نحصل على تمثيل في النظام الثماني (بقاعدة 8) ، للتشكيلة الثنائية الداخلية في حيِّز من الذاكرة مُخصَّص إلى المتحولة التي نطبعها حسب مواصفة النسق هذه .

9.6.4 ـ تكرار المواصفات

رأينا (الفقرة 9.6.1) أنه من الممكن إستباق المواصفات الخاصة بمُعامل تكرار عبارة عن ثابتة صحيحة بدون إشارة .

بنفس الطريقة ، من الممكن إستباق مجموعة مواصفات بمُعامِل تكرار شبيه . في هذه الحالة ، توضع مجموعة المواصفات بين أهلَّة .

مثلاً : الكتابة التالية :

```
154 FORMAT(1H0,2(3X,I5,F10.2),I3)
```

يعادل النسق التالي :

```
154 FORMAT(1H0,3X,I5,F10.2,3X,I5,
                         F10.2,I3)
```

9.6.5 ـ إعادة إستكشاف النسق

قد يحدث بأن شكلاً أو نسقاً مناسباً لأمر بالقراءة ـ الكتابة يحتوي على مواصفات معالجة معطيات أقل من المتحولات في لائحة الادخال ـ الإخراج .

في هذه الحالة ، سيتم إعادة إستكشاف النسق من بدايته .

مثلاً :

```
    WRITE(6,212) A,B,C,D,E,F
212 FORMAT(2X,4F12.3)
```

هذه التعليمات سيتم تنفيذها كالتعليمات التالية :

```
    WRITE(6,212) A,B,C,D,E,F
213 FORMAT(2X,4F12.3,2X,2F12.3)
                     ↑
             نتيجة الاستكشاف
```

9.6.6 ـ حالة خاصة بالطابعات السريعة

في الحالة الخاصة بالطابعات السريعة ، يجب أن نتذكّر إن سمة التسجيل الأولى تستخدم كسمة تحكّم وطفور ولا تطبع أبداً .

قواعد التحكّم والطفور توجز على الشكل التالي :

مهمة	سمة رقم 1
طفور عن صفحة	1
مساحة بين الأسطر مزدوجة	0
مساحة بين الأسطر مثلثة	-
إلغاء الطفور	+
ـ مساحة ما بين الأسطر عادية	بياض
مساحة ما بين الأسطر بسيطة (*)	سمات مختلفة

(*) نقفز إلى « القنال » المناسب لبرنامج التحكّم .

10 ـ البرامج الثانوية

هناك أربعة أنواع من البرامج الثانوية في لغة FORTRAN : الصِيغ والدوال ، المناهج الثانوية والمجموعات BLOCK DATA .

10.1 ـ الصيغ الرياضية

تعرَّف الدالة « صيغة رياضية formule » في بداية البرنامج بواسطة تعبير خاص بلغة فورتران .

أمثلة :

```
FL1(X) = (1./A) * EXP(-A/X/X)
TG(Y) = SIN(Y)/COS(Y)
```

Y وX الموجودة على يسار الاشارة = ، هي عبارة عن متحولات خرساء : يُمكن أن تُستبدل بواسطة أي تعبير جبري عند الاستعمال الداخلي للدالة المعرَّفة .

في المثل (1) ، المتحولة A هي على العكس متحولة حقيقية : فهي غير موجودة على يسار الاشارة = . يجب أن تكون معرَّفة قبل أي طلب للدالة FL1(X) .

القواعد :

أ ـ يجب أن تكون الصيغة الرياضية في بداية البرنامج ، وقبل أي تعليمة قابلة للتنفيذ .

ب ـ عند الاقتضاء ، يجب أن يُصرَّح عن نوع الدالة قبل كتابة الصيغة (يُمكن أن تكون فعلياً بسيطة أو بدقة مزدوجة ، صحيحة ، منطقية أو مركبة) .

ج ـ لا يُمكن أن يكون هناك سوى صيغة واحدة لدالة معينة .

د ـ كل موضوع من اللغة : ثوابت ، متحولات من جميع الأنواع ، متحولات لا توجيهية (Scalar) أو متجهات ، دالات أخرى ، يُمكن أن تكون موجودة في الدالة .

هـ ـ التكرار هو ممنوع : إسم الدالة التي نحدِّدها لا يمكن أن يكون موجوداً على يمين الإشارة = .

10.2 ـ الدالات

الدالة فورتران هي عبارة عن برنامج ثانوي حقيقي ، نكتبها بشكل عام بعد الأمر END في البرنامج الرئيسي . وهي تبدأ بالأمر FUNCTION ، وتنتهي بالأمر END . ويجب أن تحتوي على الأقل على أمر RETURN .

الشكل العام للدالة فورتران هو :

```
FUNCTION nom(liste)
.
.
END
```

nom هو إسم الدالة . ويخضع للقواعد العادية المناسبة لأسماء المتحولات .
liste عبارة عن لائحة من المتغيرات الوسيطية ، المتحولات الصامتة الموجـودة في نفس الوقت في هذه اللائحة وفي جسم الدالة .

مثلاً :

```
      FUNCTION FACT(X)
      IF(X.EQ.0.) GO TO 10
      IF(X.LT.0.) GO TO 11
      FACT = 1.
      KMAX = X
      DO 12 K = 2, KMAX
   12 FACT = FACT * FLOAT(K)
      RETURN
   10 FACT = 1.
      RETURN
   11 WRITE(6,100)
  100 FORMAT(/2X, 'ERREUR,X NEGATIF')
      RETURN
      END
```

هـذا البرنامج يحسب الدالة العـاملية (Foctoricile) للقسم الصحيح لأي تعبير جبري . ومن الطبيعي أن يكون هناك عدة طرق لكتابته .

للإشارة : القيمة المحسـوبـة بالدالة تُعاد الى البرنامج الرئيسي بواسطة إسم الدالة (في هذا المثل الاسم هو : FACT) .

الأمر RETURN يُعيد التحكم إلى البرنامج الرئيسي ، إلى التعليمة الآلية التي تتبع مباشرة دعوة البرنامج دالة . دعوة البرنامج دالة تتم بواسطة إسمها ، وكأي دالة رياضية عادية أو دالة خدمة .

10.3 ـ البرامج الثانوية (المنهاج الثانوي SUBROUTINE)

يبنى البرنامج الثانوي كدالة من نوع FUNCTION . وفي أغلب الأحيان يختلف الأمر الأول وطريقة الطلب .

الشكل العام :

```
SUBROUTINE nom(liste)
.
.
.
END
```

nom وListe هي شبيهة بالدوال من نوع FUNCTION يجب أن يكون هناك أمر من نوع RETURN على الأقل في جسم البرنامج الثانوي .

دعوة البرنامج الثانوي SUBROUTINE هي ضمنية : وتتم بواسطة أمر من نوع CALL .

مثلاً :

```
CALL TRI1(TAB,N)
```

كما في الدوال من نوع FUNCTION ، كل تعليمة فورتران هي صالحة للاستعمال في برنامج ثانوي من نوع SUBROUTINE .

10.4 ـ القواعد المناسبة للمتحولات

المتحولات الموجودة في برنامج ـ ثانوي من نوع دالة أو منهاج ثانوي SUBROUTINE هي نوعان :

أ ـ متحولات شكلية (صامتة)

هي تلك الموجودة في اللائحة وفي جسم البرنامج الثانوي . ويُمكن أن تكون من جميع الأنواع ، لا إتجاهية (Scalaire) أو غيرها . وعند الإقتضاء ، يجب أن يكون مصرّحاً عنها في بداية البرنامج ـ الثانوي ، بشكل متكيِّف مع مقاييس (argument) لائحة الطلب .

ب ـ المتحولات العادية

تكون موجودة في جسم البرنامج ، وليس في اللائحة .

ج ـ التعرُّف على المتحولات

متحولات البرنامج ـ الثانوي ، الصامتة وغير الصامتة (الشكلية والحقيقية) ، لا تعرف إلا في البرنامج الثانوي موضوع السؤال . هكذا ، مثلاً ، فالمتحولة ALPHA المستعملة في برنامج رئيسي لا يوجد أي شيء مشترك بينها وبين متحولة بنفس الاسم ALPHA مستقلة في البرامج الثانوية ومدعوة بواسطة برنامج رئيسي . نفس الشيء بالنسبة لأوسمة التعليمات .

10.5 ـ المجموعة BLOCK DATA

يتعلَّق ذلك ببرنامج ثانوي خاص ، موجه لاعداد المتحولات عند التصريف . ولا يحتوي على أية تعليمة قابلة للتنفيذ ، ولكن فقط على تعليمات للتصريح .

الدور الوحيد لبرنامج ثانوي من نوع BLOCK DATA هو إعداد متحولات مصرَّح عنها بشكلٍ مشترك في داخله عند التصريف .

مثلاً :

```
BLOCK DATA
COMMON X,Y,Z,K
DATA X,Y,Z,K/1.5,2.,0.,27/
END
```

11 ـ الأوامر COMMON وEQUIVALENCE

EQUIVALENCE ـ 11.1

تسمح هذه التعليمة للمصرِّف بتخصيص نفس الذاكرة إلى عدة متحولات .

الشكل العام :

EQUIVALENCE (liste 1), (liste 2), ..., (liste *n*)

اللائحة Liste i هي عبارة عن لائحة بالمتحولات اللاتوجيهية (Scalar) أو بالمتحولات المؤشرة بواسطة ثوابت.

مثلاً :

```
EQUIVALENCE(X,Y,K),(TAB(1),M(5))
```

المتحولات الحقيقية X وY إضافة إلى المتحولة K لها نفس العنوان الفيزيائي في الذاكرة .

تُحسب عناوين الجداول TAB (الحقيقية) و M (الصحيحة) بشكل يكون فيه عنوان (1) TAB شبيه (يُساوي) عنوان M(5) . هذا العنوان يختلف بالتأكيد عن العنوان المشترك لـ X, Y وK .

COMMON ـ 11.2

هدف هذه التعليمة هو تخصيص عناوين لبعض المتحولات بحيث يُمكن أن تستعمل هذه المتحولات مشتركة بواسطة برنامج أو برامج ثانوية . من الممكن أيضاً تفادي لوائح كبيرة من المُعاملات عند طلب البرامج الثانوية .

الشكل العام :

COMMON liste

liste هي عبارة عن لائحة من المتحولات اللا توجيهية . في حالة الجداول من الممكن ألا نستعمل الأبعاد . والجدول يجب أن يكون مصرّحاً عنه سابقاً في أمر من نوع DIMENSION . ولكن من الممكن تجميع نوعين من التصريحات في نوع واحد .

أمثلة :

```
COMMON A,B,TAB(20)
```

TAB هو جدول حقيقي ببعد واحد ويتألف من 20 عنصراً .
الاستعمال العادي للتعليمة COMMON يعني أن هذه التعليمة موجودة في نفس الوقت في البرنامج الرئيسي وفي البرامج الثانوية . يجب إذاً أن نتذكر أنه فقط ترتيب المتحولات في اللائحة يسمح بتعريفها ، كما هو داخلياً في لوائح المتغيرات الوسيطية والمعاملات .

وفي النهاية ، يجب على المبرمج الانتباه إلى تفاعل التعليمات EQUIVALENCE وCOMMON .

12 ـ توسيعات اللغة

12.1 ـ متحولات من نوع سمات

حمل التوسيع FORTRAN 77 إلى اللغة قسماً من سهولة اللغة PL / 1 في معالجة سلاسل السمات .

التصريح يتم على الشكل التالي :
الشكل العام :

```
CHARACTER * n liste
```

حيث Liste هي لائحة بالمتحولات اللاتوجيهية أو البعدية . n هي الطول الثابت لسلاسل السمات المسماة في اللائحة ؛ n هي ثابتة صحيحة إيجابية .

أمثلة :

```
CHARACTER * 34 A1, A2, B, X
CHARACTER * 12 TAB (20)
CHARACTER *  5 REP, * 8 ALPHA, BRAVO
```

يوجد في FORTRAN 77 تعليمات وعمليات لمعالجة سلاسل السمات : ويُنصح القارىء بالعودة إلى الملاحظات حول المواصفات التقنية للمصرِّف الذي يستعمله من أجل الحصول على تفصيلات أكثر ، هذه العمليات ليست مغايرة حتى الآن .

12.3 ـ التحويلات الداخلية

O الأمر DECODE (أنظر الأمر READ) .
الشكل العام هو :

```
DECODE(X, f) liste
```

يوجد « قراءة » لسلسلة السمات X ، حسب النسق f ، وإرسال المعلومات ، مع تحويل ، إلى عناصر اللائحة liste .

مثلاً :

```
   DECODE(X, 12) K, L
12 FORMAT(I2, 1X, I3)
```

● الأمر ENCODE (أنظر الأمر WRITE) .

الشكل العام :

```
ENCODE(X, f) liste
```

هناك « كتابة » لقيم عناصر اللائحة ، مع تحويل ، في سلسلة السمات X حسب النسق f .

مثلاً :

```
   ENCODE(X, 15) A, J
15 FORMAT(2X, F12.3, I8)
```

مراجع

كتب أساسية

الوثائق التقنية للمُصمّمين ، والمناسبة للمصرّفات فورتران المستعملة .

من الممكن أيضاً إستشارة :

M. DREYFUS, *FORTRAN IV*, Dunod (réédition 1982).



الفصـــل السـابــع

كوبول COBOL

1 ـ مواضيع اللغة

1.1 ـ مجموعة السمات

تستعمل لغة كوبول مجموعة من السمات تعادل 51 سمـة : البياض : للبيـاض (الفراغ) دور مهـم في لغة كوبول. فخارج المعطيات الأبجعددية ، يستخدم كفاصـل إلزامي لمختلف المعرّفات أو مؤثرات هذه اللغة .

للإشارة : فقط في المعلومات الأبجعددية ، وفي كل مرة نستعمل فيها فراغ ، يمكننا وضع ما نرغب منها .

الأحـرف : هي السمات الأبجـدية ، من A إلى Z . وتمثيلهـا دائـماً بـالأحـرف الكبيرة . في هذا الفصل ، وكما في الفصل السابق ، الأحرف ليست قضيبية .

الأرقام : هي السمات الرقمية العشر من 0 إلى 9 .

والسمات الخاصة هي التالية :

+ زائد	؛ نقطة فاصلة
- ناقص	(هلال مفتوح ، أو يسار
* نجمة	) هلال مغلق ، أو يمين
/ قضيب منحرف (slash)	< أصغر من
= يعادل	> أكبر من
. نقطة	$ دولار
, فاصلة	' أبوستروف أو «quote» .

1.2 ـ الثوابت الحرفية

1.2.1 ـ الثوابت الرقمية

تمثيل الثابتة الرقمية يتم بواسطة سلسلة من السمات المؤلفة من إشارة ، أرقام وعند الاقتضاء من فاصلة عشرية .

مثلاً :

```
   12
-  14.544
+  250
    0
```

الخ

قواعد :

أ ـ العدد الكامل للسمات الرقمية يجب أن يكون أقل أو يعادل 18

ب ـ غياب الإشارة يعني قيمة إيجابية .

ج ـ عندما تكون الإشارة + أو – موجودة ، فيجب أن تسبق مباشرة الرقم الأول ، بدون فراغ أو بياض وسيط .

د ـ عندما نستعمل الفاصلة العشرية ، فيجب أن تكون مسبوقة مباشرة برقم : أو بكلمة أخرى ، ثابتة رقمية صحيحة لا تحتوي أبداً على فاصلة عشرية لجهة اليمين .

1.2.2 ـ ثوابت غير رقمية

هي عبارة عن سلاسل السمات في المعنى العام . وتتألف عادة من سلسلة من السمات المختلفة محصورة بين أبوستروف ('') ، الطول الأقصى للثابتة غير الرقمية يتعلّق بالنظام المستعمل .

أمثلة :

```
'JULES'
'253.3.4 + 2'
'JEAN-PIERRE' etc...
```

قواعد :

أ ـ تحتوي سلسلة السمات على أي سمة مهما تكن ما عدا أبوستروف (') .

ب ـ الطول الأقصى لسلسلة السمات يتعلّق بنوع المصرّف المستعمل .

1.2.3 ـ الثوابت التصويرية (الرسمية)

الثوابت التصويرية هي من مميزات لغة كوبول ، وهي عبارة عن معرّفات بقيمة ثابتة .

Converted by Tiff Combine - (no stamps are applied by registered version)

جدول الثوابت التصويرية

قيمة	ثوابت تصويرية
ZERO	ZERO أو ZEROS أو ZEROES
بياض	SPACE أو SPACES
سمة بقيمة ثنائية قصوى	HIGH-VALUE أو HIGH-VALUES
سمة بقيمة ثنائية دنيا	LOW-VALUE أو LOW-VALUES
أبوستروف (")	QUOTE أو QUOTES

الكلمة ALL التي تسبق ثابتة تصويرية تولد سلسلة من السمات متشابهة مع الثابتة .

1.3 ـ المتحولات

1.3.1 ـ قواعد كتابة الأسماء بلغة كوبول

المتحولات بلغة كوبول ، تتبع القواعد العامة لكتابة الأسماء في اللغة . وهي تتمثّل بواسطة سلسلة من السمات الأبجدية أو الرقمية أو الأبجعددية . من الممكن أيضاً إستعمال السمة ـ (ناقص) بشرط أن لا تكون الأولى ولا الأخيرة في الاسم ، وأن تكون مسبوقة ومتبوعة بسمة أخرى . الطول الأقصى للاسم هو 30 سمة .

أمثلة :

```
ALPHA
NOM-DOSSIER
X3445
9AB76
```

ملاحظة :

أ ـ السمة الأولى يُمكن أن تكون رقماً .

ب ـ يجب أن يكون هناك دائماً سمة غير رقمية في الاسم .

1.3.2 ـ تركيبة المتحولات

المتحولات يمكن أن تكون لا توجيهية (Scalar) ، تراتبية (عشيرية hierarchisé ، أو مؤلفة من جداول .

تفصيل المتحولات سيتم في الفقرات المناسبة للقسم DATA DIVISION .

يُصرّح عن المتحولات في لغة الكوبول دائماً بشكل واضح .

1.3.3 ـ أوامر اللغة

أوامر اللغة تدعى أيضاً أفعال Verbes كوبول .

كتابة الأوامر تخضع أيضاً لقواعد كتابة الاسماء .

جميع أفعال الكوبول ، إضافة إلى الثوابت التصويرية وبعض الأسماء في اللغة ، هي كلمات محفوظة . ولا يمكن إستعمالها عادة .

لائحة كلمات كوبول المحفوظة هي موجودة في نهاية هذا الفصل ، في الصفحة (127) .

2 ـ قواعد الكتابة كوبول

2.1 ـ نسق (شكل) البطاقة

نسق (شكل) البطاقة هو التالي (*)

1 2 3 4 5 6	7	8 9 10 11	12 13 14 72	73 74 75 76 77 78 79 80
N1		A	B	N2

ـ الحيِّز N1 ، الأعمدة من 1 إلى 6 ، هي للاستعمال الاختياري . يُمكن أن تستعمل قيم الأسطر من البرنامج .

ـ العامود 7 ، يدل على تتابع محتمل : في هذه الحالة ، يحتوي على إشارة – (ناقص) (لا يُنصح باستعماله) .

ـ الحيز A ، الأعمدة من 8 إلى 11 ، يجب أن يحتوي على بعض كلمات اللغة ، إضافة إلى عدد من المستويات .

ـ الحيز B ، الأعمدة من 12 إلى 72 ، يحتوي على جميع عناصر البرنامج التي لا تُسجَّل في الحيز B .

ـ الحيز N2 ، الأعمدة 73 إلى 80 ، هو كما في جميع لغات البرمجة ، مهمل من قِبل المصرِّف .

(*) الاستعراض الجبري على البطاقة يبقى صالحاً بالنسبة للأداة الطرفية (terminal) .

2.2 ـ قواعد الترقيم

لبعض السمات الخاصة ترقيم . وهي التالية :

. نقطة ترقيم إجباري نهاية الجملة

، فاصلة إختياري (لا ينصح بها) .

؛ نقطة ـ فاصلة إختياري (لا ينصح بها) .

(هلال مفتوح بداية اللائحة

) هلال إغلاق نهاية اللائحة

القواعد :

أ ـ لا يجب أن تكون نقطة الترقيم مسبوقة ببياض كما ويجب أن تكون دائماً متبوعة بسمة بياض واحدة على الأقل (بهذه الطريقة يستطيع المصرِّف تفرقتها عن الفاصلة العشرية) .

ب ـ نفس القواعد بالنسبة للفاصلة والفاصلة العشرية .

ج ـ الهلال المفتوح هو إلزامي مسبوق بسمة بياض واحدة على الأقل .

د ـ الهلال المغلق يجب أن يكون متبوعاً بسمة بياض واحدة على الأقل ، ما عدا في نهاية الجملة .

ملاحظة : هناك حالة تكون فيها الفاصلة إلزامية : بين دلائل (مؤشر indice) المتحولة المصرَّح عنها في جدول بعدة أبعاد .

مثلاً :

```
MOVE ZERO TO TABLE(K1, K2)
```

السمات الخاصة المناسبة للعمليات الجبرية أو المقارنة (أي + ، - ، * ، / ، < ، = ، و >) . يجب أن تكون ، وفي جميع الحالات ، متبوعة ، ومسبوقة بسمة بياض واحدة على الأقل .

3 ـ التركيبة العامة للبرنامج بلغة كوبول

يحتوي البرنامج بلغة الكوبول ، وفي ترتيب معين على أربعة أقسام مختلفة ، تدعى DIVISION . يبدأ كل قسم بكلمة محجوزة متبوعة بالكلمة DIVISION . الكلمة المحجوزة في اللغة تبدأ في العامود 8 (القسم A) ، كجميع أسماء الفقرات .

كل قسم يحتوي بدوره على أقسام ثانوية ، من نوع قطاع دائري Section أو فقرة Paragraphe ، حيث البعض منها إلزامي .

التراتبية العامة objet في لغة كوبول هي التالية :

السمات

كلمات (30 سمة على الأكثر من Z ÷ A ، 0-9 و « - ») .

تعليمات

جمل (تنتهي بنقطة متبوعة بفراغ)

فقرات (إسم في الحيز A ، العامود 8) .

قطاعات (إسم في الحيز Z ، متبوع بالكلمة «SECTION») .

أقسام (أربعة أقسام إلزامية ، في ترتيب البرنامج) .

هيكل برنامج بلغة كوبول هو مُوحَّـد دائماً : أنظر الجدول 1 التالي ؛ إدارة الذاكرة بلغة كوبول هي كما في لغة فورتران من نوع ساكن . وهذا يعني ، بشكل خاص ، إن جميع المتحولات في البرنامج تنشأ قبل بداية تنفيذه وتبقى جاهزة بتصرفنا حتى عودة التحكم إلى المشرف (أمر STOP RUN) .

جدول 1 ـ هيكلية البرنامج بلغة كوبول

```
8... 12...
IDENTIFICATION DIVISION.
PROGRAM-ID. XXXXXXXX.
REMARKS. CECI N'EST PAS UN PROGRAMME
 COMPLET, MAIS SEULEMENT UN SQUELETTE
 DE PROGRAMME. SEULES SONT INDIQUEES
 LES SUBDIVISIONS OBLIGATOIRES.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. YYYYYYYYYYY.
OBJECT-COMPUTER. YYYYYYYYYYY.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
     SELECT ZZZZZ ASSIGN TO.....
     ...
DATA DIVISION.
FILE SECTION.
FD ZZZZZ .....
01 XYZXY.
   02
     ...
WORKING-STORAGE SECTION.
77 ALPHA ...
   ...
PROCEDURE DIVISION.
...
...
```

4 ـ القسم IDENTIFICATION DIVISION

التركيبة العامة هي التالية :

```
IDENTIFICATION DIVISION.
PROGRAM-ID. XXXXX.
AUTHOR. YY...Y.
INSTALLATION. ZZ...Z.
DATE-WRITTEN.
DATE-COMPILED.
SECURITY.
REMARKS.
```

فقط، السطران الأولان هما إلزاميان. جميع الفقرات أو الأسطر التالية هي إختيارية وهـا قيمة الملاحظيـة (COMMENTAIRE) . وفي بعض الحالات ، التي تقـوم فيها باستعمالها يجب أن تكون في هذا الترتيب .

لا يجب أن يزيد إسم البرنامج ، الذي يتبع PROGRAM-ID ، عن 8 سمات (أو 6 ، على الأكثر ، حسب المصرِّف) . يُقبل الطول من 6 سمات من أغلبية الأنظمة .

القسم IDENTIFICATION ليس له أية مهمة في لغة كوبول .

ENVIRONMENT DIVISION ـ 5

التركيبة العامة لهذا القسم هي التالية :

```
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER.
OBJECT-COMPUTER.
SPECIAL-NAMES.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
I-O-CONTROL.
```

5.1 ـ القسم CONFIGURATION

السطران أو الفقرتان الأولتان لهما قيمة الملاحظية .

مثلاً :

```
SOURCE-COMPUTER. XXXXXX.
OBJECT-COMPUTER. YYYYYY.
```

يُمكن أن تهمل ، ككامل القسم CONFIGURATION

ملاحظة : في COBOL A.N.S.I، القسم CONFIGURATION هـو إلزامي ،

إضافة إلى الفقرتين الأولتين(*) .

الفقرة الثالثة (SPECIAL-NAMES) تسمح بإعادة تعريف بعض المواضيع في اللغة :

```
SPECIAL-NAMES.
syst-nom IS nom
CURRENCY SIGN IS littéral
DECIMAL-POINT IS COMMA.
```

الجملة الأولى تعرِّف إسم النظام nom-système ، التي يُمكن أن تكون وحدة إدخال - إخراج (مثلاً : Sysout, SYSPUNCH,) . الإسم nom ، وراء IS هو باختيار المبرمج .

الجملة الثانية تعرِّف سمة العملة .

الجملة الثالثة تسمح بعكس أدوار الفاصلة والنقطة العشرية .

5.2 ـ القسم INPUT-OUTPUT

5.2.1 ـ الفقرة FILE-CONTROL

```
FILE-CONTROL.
SELECT nom-fichier
ASSIGN TO nom-système
ACCESS MODE IS SEQUENTIAL/RANDOM
ACTUAL KEY IS nom-donnée.
```

nom-fichier هو عبارة عن إسم يختاره المبرمج ، وسيتم مراجعته في القسمين التاليين .

nom-système هو إسم وحدة إدخال - إخراج ، مُحدَّد بواسطة المُصمِّم .

الجملة SELECT ... ASSIGN TO... تسمح بإجراء رباط بين السجل الرمزي ، الذي لا وجود له إلا في حدود البرنامج ، والناقل الفيزيائي الذي ، هو دائم بالنسبة له .

الجملة ACCESS MODE هي غير إلزامية إلا بالنسبة للسجلات بالبلوغ المباشر . في هذه الحالة ، الجملة ACTUAL KEY هي أيضاً إلزامية . وإستعمالها ، غير مُعيَّر ويتعلَّق بالأنظمة المستعملة .

5.2.2 ـ الفقرة I-O-CONTROL

```
I-O-CONTROL.
    RERUN ON nom-système
    SAME AREA
    MULTIPLE FILE TAPE
```

(*) A.N.S.I. : American National Standards Institute.

هذه الفقرة هي إختيارية . وتتعلّق كثيراً بالنظام المستعمل . ومهمتها مراقبة حيزات الدارىء في الإدخال - الإخراج (buffers) ، والتحكّم بنقاط العودة المحتملة إضافة إلى استعمال السجلات المتعددة على الأشرطة المغناطيسية .

ولتفصيلات أكثر يُنصح القارىء بالرجوع إلى الوثائق التابعة للنظام .

DIVISION DATA - 6

التركيبة العامة هي :

```
DATA DIVISION.
FILE SECTION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.(note : section non au COBOL
                 officiel)
REPORT SECTION.(note : section facultative)
```

يوجد القطاعان الأولان (WORKING -STORAGE و FILE) إلزامياً على جميع الأنظمة . أما الأخرى فيمكن أن تتغيّر من نظام إلى آخر .

القسم DATA هو القسم من البرنامج الذي يتم فيه التصريح ووصف جميع المعطيات في البرنامج ، بما فيها السجلات المنطقية . الشكل العام لوصف المعطيات هو التالي :

nn nom-donnée clause 1 clause.2 ... clause n.

nn هو رقم المستوى ، ويتألف من سمتين . ويوجد عند بداية جميع أنواع وصف المعطيات ، إما في القسم A وإما في القسم B .

nom-donnée هو عبارة عن إسم مُختار بواسطة المبرمج . ويُمكن أن يكون الإسم FILLER ، في الحالة التي نرغب فيها فقط بحجز حيّز من المعطيات بدون إستعمال داخلي .

البند Clause i هو عبارة عن وصف للمعطيات .

النقطة ، المتبوعة ببياض واحد على الأقل ، تنهي عملية الوصف .

6.1 - عدد المستويات

الجدول 2 يختزل اللائحة ومهام « عدد » المستويات الموجودة في لغة COBOL .

الجدول 2 ـ عدد المستويات

FONCTION المهمة	MARGE المكان	عدد المستويات
وصف السجل المنطقي (FILE فقط)	A	FD
وصف سجل الفرز (FILE فقط)	A	SD
وصف السجل REPORT (FILE فقط)	A	RD
تركيبة	A	01
تركيبة ـ ثانوية من تركيبة أو معطيات بسيطة	A أو B	02-49
معطيات معزولة (فقط WORKING-STORAGE)	A	77
إسم الشرط	A	88
إسم التجميع (البند RENAMES)	A	66

المستويات FD ، SD وRD لا تظهر إلّا في القسم FILE SECTION . المستويات 77 لا تظهر إلا في القسم WORKING-STORAGE SECTION المستويات الأخرى يمكن أن تظهر في جميع الأقسام .

الترتيب المطلوب المحافظة عليه هو التالي :

```
FILE SECTION.
FD ...
01 ...
FD ...
01 ...
88 ...
66 ...
WORKING-STORAGE SECTION.
77 ...
77 ...
01 ...
01 ...
88 ...
66 ...
```

6.2 ـ وصف السجلات

الشكل العام :

```
FD nom-de-fichier
BLOCK CONTAINS
RECORD CONTAINS
RECORDING MODE
LABEL RECORD
DATA RECORD
```

6.2.1 ـ البند BLOCK CONTAINS

يصف البلوك الفيزيائي المؤلف للسجل

الشكل :

BLOCK CONTAINS entier1[TO entier2] { CHARACTERS / RECORDS }

6.2.2 ـ البند RECORD CONTAINS

يُستعمل لوصف الحجم ، المصرّح عنه دائماً بعدد السمات ، لتسجيلة منطقية .

الشكل :

RECORD CONTAINS entier1 [TO entier2] CHARACTERS

entier 1, entier 2 ـ عبارة عن أعداد صحيحة تؤلّف حجم التسجيلة .

6.2.3 ـ البند RECORDING MODE

يوصف طريقة التسجيل .

الشكل :

RECORDING MODE IS option

option هي كلمة محجوزة خاصة بالنظام المستعمل :

6.2.4 ـ البند LABEL RECORD

ويدل على الوجود المحتمل لأوسمة السجلات . هذا البند لا يوجد دائماً في وصف السجلات النموذجية للادخال ـ الاخراج (قارىء البطاقات ، مثقبات البطاقات أو لطابعات) .

الشكل :

LABEL RECORDS ARE { OMITTED / STANDARD }

6.2.5 ـ البند DATA RECORD

يربط وصف السجل المنطقي بوصف المعطيات ، وهو دائماً بمستوى 1 ، مناسب لتسجيلات هذا السجل .

الشكل :

DATA { RECORD IS / RECORDS ARE } nom1 [nom2] ...

nom 1, ... بمستوى D1 يتبع مباشرة وصف السجل .

6.3 ـ وصف المتحولات

الشكل العام :

```
nn nom-donnée ou FILLER

REDEFINES
BLANK WHEN ZERO
JUSTIFIED RIGHT
OCCURS
PICTURE
SYNCHRONIZED
USAGE IS
VALUE IS
```

حيث nn يُمكن أن تأخذ القيم من 01 إلى 49 ، أو 77 ، أو 88 أو 66 .

6.3.1 ـ البند REDEFINES

يسمح باعادة تعريف حيِّز من الذاكرة (يشبه EQUIVALENCE في لغة فورتران أو DEFINED في PL / 1) .

الشكل :

```
nom1 REDEFINES nom2
```

6.3.2 ـ البند BLANK WHEN ZERO

هذا البند يُطبَّق على المعطيات الرقمية . وعندما تكون قيمة المعطى صفراً، فكامل الحيز المناسب سيوضع في بياض .

6.33 ـ البند JUSTIFIED RIGHT

هذا البند يطبَّق على المعطيات من نوع سلسلة من السمات . وعلى عكس القاعدة الكلاسيكية للتركيز لجهة اليسار ، فإن إرسال المعطيات من حيِّز الى حيِّز آخر مع البند JUSTIFIED RIGHT يتم من خلال سمات من اليمين ويتركز لجهة اليمين .

6.3.4 ـ البند OCCURS

هذا البند يسمح بإنشاء جداول .

الشكل :

```
OCCURS e1 TO e2 TIMES DEPENDING
ON nom1
```

e1 وe2 هي عبارة عن أعداد صحيحة بدون إشارة .

nom 1 ، الذي يجب أن يكون محدّداً سابقاً، يسمح بتركيز حجم الجدول ، بين القيمتين الطرفيتين e1 وe2 .

القاعدة :

لا يمكننا الذهاب بعيداً عن الأبعاد الثلاثة في لغة كوبول . إضافة لذلك ، فالمعطيات بالمستوى 01 لا يُمكن أن تحتوي على البند OCCURS ، ما عدا المعطيات بالمستويات 66 ,77 و88 .

6.3.5 ـ البند SYNCHRONIZED

هذا البند يسمح بتسطير بداية الحيِّز المناسب إلى المتحولة على جبهة من الكلمات في الذاكرة . ويؤدي ذلك عند تطبيقه إلى نتائج مختلفة حسب المكنات المستعملة .

6.3.6 ـ البند USAGE

هذا البند يُعرِّف نوع التمثيل الداخلي الذي سيستعمل للمتحولة .

الشكل :

```
           { DISPLAY          }
           { COMPUTATIONAL    }
USAGE IS   { COMPUTATIONAL-1  }
           { COMPUTATIONAL-2  }
           { COMPUTATIONAL-3  }
```

ملاحظة : الكلمة COMPUTATIONAL يُمكن أن تختصر بالكلمة COMP .

الجدول 3 ـ يعطي التناسب بين هذه الأوصاف والتمثيل الداخلي ، في الحالة التي تكون فيها المكنة ببايتة واحدة .

جدول 3 ـ حالة مكنة ببايتة واحدة

OPTION	CLASSE	REPRESENTATION
DISPLAY	أبجعددي أو رقمي مُنقّح	سمات (صيغة ضمنية بالغلط تُؤخذ دائماً في حالة غياب البند USAGE)
COMP	رقمي	ثنائي صحيح (على 2 ,4 أو 8 بايتات)
COMP-1	رقمي	بفاصلة متحركة وبدقة بسيطة مُعيَّرة (على 4 بايتات)
COMP-2	رقمي	بفاصلة متحركة وبدقة مزدوجة معيَّرة على 8 بايتات)
COMP-3	رقمي	عشري مكثّف : رقمي في كل بايتة

6.3.7 ـ البند VALUE

هذا البند يسمح بتصفير المتحولة .

الشكل :

```
VALUE IS littéral
```

القواعد :

البنـد VALUE لا يستعمـل إلا في القسم WORKING-STORAGE . ومن الممكن في أغلب الأحيان إستعمال القسم FILE ، ولكن فقط في الحالة التي تكون فيها المتحولات بمستوى 88 (أسماء ـ شرطية) . من الأفضل وضع هذا البند في الموقع الأخير خلف جميع البنود المناسبة لمتحولة معينة .

6.3.8 ـ البند PICTURE

هذا البند ، هو مهم للغاية ، يُوصف بواسطة أكواد ـ سمات خاصة ، الصـورة الخارجية لمتحولة معينة .

الشكل :

```
PICTURE IS chaine
```

حيث Chaine هي سلسلة من أكواد ـ سمات الوصف .

أمثلة :

```
PICTURE IS XXXXX.
PICTURE IS 999V99.
PICTURE IS A(15).
```

مُعامِل التكرار :

من الممكن إستعمال مُعامِل تكرار ، هو عبارة عن ثابتة بين هلالين تتبع مبـاشرة السمة المطلوبة . هكذا ، X(5) تعادل XXXXX .

هنـاك ثلاث طبقـات من PICTURE : الأوصـاف الأبجعـددية ، الأبجـديـة والرقمية . من هذه الأبجعددية نستطيع تمييز سلاسل السمات العادية ، وتلك التي تمتاز بمعنى رقمي : وهي عبارة عن الصورة PICTURE الرقمية المُنقَّحة .

أ ـ الصورة الأبجدية PICTURE alphabetic

code : [A]

كل موقع مُعلَّم بواسطة A لا يُمكن أن يحتوي إلا على حرف أو فراغ .

مثلاً :

```
PICTURE AAAAAA.
```

Converted by Tiff Combine - (no stamps are applied by registered version)

ب ـ الصورة الأبجعددية PICTURE Alphanumeric

codes : | A, X |

كل موقع مُعلَّـم (يوجد فيه) بواسطة X يُمكن أن يحتوي على أية سمة .

مثلاً :

```
PICTURE XXXXAA.
```

ج ـ الصورة الرقمية

codes ; | 9,V,S |

كل موقع يحتوي (معلَّـم) على 9 لا يُمكن أن يحتوي إلا على رقم الكود ـ السمة V يدل على موقع تـركيز (alignement) الفـاصلة العشريـة ، ولكنه لا يُحـدِّد أبداً إدخـال الفاصلة .

الكود ـ السمة S ، دائماً وحيد ويوضع في رأس السلسلة ويدل على موقع وجـود الإشارة الحسابية للعدد المحدَّد .

مثلاً :

```
PICTURE S999V99.
```

د ـ الصورة الرقمية المنقحة PICTURE numeric-edited

الكود : | B 0 . , Z * $ + - |

السمات التالية :

B تؤدي إلى إدخال بياض (فراغ) .

0 تؤدي إلى إدخال صفر .

. تؤدي إلى إدخال نقطة (.) وتفرض في هذه النقطة تركيز الفاصلة العشرية . ويجب أن تكون وحيدة في السلسلة .

' تؤدي إلى إدخال فاصلة .

Z تستعمل بدلاً عن 9 . وإذا كان الرقم الواقع في هذا الموقع هو 0 بدون دلالة ، فسيتم إبداله عند التنقيح بواسطة بياض .

* تعمل مثل Z ، مع الفارق في إن 0 بدون دلالة سيُستبدل بواسطة السمة نجمة .

+ هذه السمة يُمكن أن تكون موجودة إما في بداية ، وإما في نهاية السلسلة PICTURE . وتؤدي إلى طباعة الاشارة (+ أو −) للعدد .

ـ تعمل مثل السمة + ، مع الفارق في أنه سيجري إخراج أو طباعة بياض (فراغ) ، إذا كان العدد إيجابياً .

$ السمة النقدية يُمكن أن تكون موجودة في رأس السلسلة PICTURE : وتؤدي إلى إخراج السمة S .

هـ ـ أكواد السمات « الزاحلة » + و − .

يُمكن أن تبدأ السلسلة PICTURE بواسطة عدة سمات + أو − متتالية : الأصفار (0) بدون دلالة الموجودة في هذه المواقع ستُستبدل بواسطة فراغات ، ما عدا 0 بـدون دلالة والموجود في أقصى اليمين ، والذي سيُستبدل بواسطة الإشارة + أو −، حسب القواعد المناسبة لهذه الأكواد ـ سمات .

ملاحظة : القواعد PICTURE COBOL هي كثيرة الشبه بتلك المستعملة في لغة PL/ 1 : الفوارق تتعلّـق بمعامِل التكرار ، النقطة والوضع بداخل أبوستروف .

7 ـ القسم DIVISION PROCEDURE

هذا القسم الأخير يؤلف البرنامج كوبول . وهو يحتـوي على قـدر ما نـرغب من الفقرات والأقسام . وفي الحد الأدنى ، هو لا يحتوي إلا على فقرة واحدة . هكذا ، فالوسم في الفقرة أو من القسم يجب أن يتبع مباشرة لقب القسم .

7.1 ـ تعليمات معالجة المعطيات

7.11 ـ MOVE

هي تعليمة إرسال ، من حيّـز إلى آخر .

الشكل 1 :

```
MOVE nom 1 TO nom 2 [nom 3] ...
```

nom1 ، يُمكن أن تكون متحولة أو تشكيلة حرفية (من الأحـرف) .

nom 2, nom 3, ... عبارة عن أسماء متحولات .

الشكل 2 :

```
CORRESPONDING nom 4 TO nom 5
```

nom 5, nom 4 ، هي أسماء تركيبات . عملية الإرسال لا تتم إذاً إلا لتركيبات ـ ثانوية أو عناصر تركيبات nom 4 وnom 5 بنفس الأسماء .

EXAMINE 7.1.2

الشكل 1 :

```
EXAMINE nom TALLYING { UNTIL FIRST } lit1
                     {     ALL     }
                     {   LEADING   }
[REPLACING BY lit2]
```

سيجري عملية كنس للسلسلة nom مع تغيير المرصف TALLY (يعادل متحولة (PICTURE 9 (5)

أ ـ الصيغة **UNTIL FIRST** : المرصف TALLY يحتوي على عدد السمات السابقة من اليسار لأول مصادفة للسمة Lit 1 (litteral) .

ب ـ الصيغة ALL : المرصف TALLY يحتوي على عدد المرّات التي يُصادف فيها السمة Lit 1 في nom .

ج ـ الصيغة LEADING : المرصف TALLY يحتوي على عدد السمات Lit 1 السابقة من اليسار لأول سمة في nom والتي تختلف عن Lit 1 .

د ـ في الحالات الثلاث، إذا كانت الصيغة **REPLACING BY** حاضرة ، فكل سمة Lit 1 ستُبدَّل بواسطة Lit 2 (الحالة UNTIL FIRST) .

الشكل 2 :

```
EXAMINE nom REPLACING { ALL         } lit1 BY lit2
                      { LEADING     }
                      { FIRST       }
                      { UNTIL FIRST }
```

طريقة عمل التعليمة بهذا الشكل هي شبيهة بالحالة الأولى ، إلى هذا تقريباً المرصف TALLY غير معدَّل .

7.2 ـ التعليمات الجبرية

الشكل :

```
instruction 1 ON SIZE ERROR ins 2, ins 3.
```

instruction 1 هي تعليمة جبرية .

ins 3, ins 2 هي عبارة عن تعليمات مختلفة . من الممكن أن يكون موجوداً منها عدد مختلف ، النقطة في النهاية تغلق الجملة .

التشغيل : في حالة الخطأ في الحجم (الحيِّز المُستقبِل غير كاف)، سيتم تنفيذ مخرج

التعليمات ins 2 ، ins 2 ، ins 3 (الخ . . .) . وإلا ، سينتقل التحكّم إلى التعليمات التالية مباشرة لنقطة نهاية الجملة .

7.2.2 ـ تدوير ROUNDED

هذه الصيغة ، التي تتبع مباشرةٍ إسم ـ المعطى (nom-donnée) لنتيجة العملية ، تؤدي إلى الحساب الأوتوماتيكي لتدوير على آخر عشر محدَّد . هذه التعليمة هي مفيـدة بشكل ملحوظ وينصح باستعمالها بانتظام .

7.2.3 ـ الجمع : ADD

الشكل 1 :

<u>ADD</u> nom 1 [nom 2] ... nom-*n* [<u>ROUNDED</u>]

nom 1, nom 2, ... هي عبارة عن أسماء متحولات أو متحولات حرفية .

nom -n هو إسم المتحولة التي ستحتوي على المجموع :

nom-1 + nom-2 + . . . + nom-n

الشكل 2 :

<u>ADD</u> nom 1 [nom 2] ... *TO* nom-*n* [<u>ROUNDED</u>]

نفس طريقة عمل وتعريف الشكل 1 .

الشكل 3 :

<u>ADD</u> nom 1 [nom 2] ... <u>GIVING</u> nom-*n* [<u>ROUNDED</u>]

nom2, nom 1 . . . هي عبارة عن أسماء متحولات أو متحولات أو ثوابت صرفية .

nom-n هو إسم المتحولة التي ستحتوي على المجمع (ما عدا + nom-n) . . .

nom1 + nom2

7.2.4 ـ الطرح : SUBTRACT

الشكل 1 :

<u>SUBTRACT</u> nom 1 [nom 2] ... <u>FROM</u> nom-*n* [<u>ROUNDED</u>.

nom2, nom1 . . . هي عبارة عن أسماء متحولات أو ثوابت حرفية (literal) .

nom-n هو إسم المتحولة التي ستحتوي على الفرق .

nom-n — nom1 — nom2 — ...

الشكل 2 :

SUBTRACT nom 1 [nom 2] ... FROM nom-p
GIVING nom-n [ROUNDED]

المتحولة nom-n هي مشحونة بواسطة نتيجة العملية :

nom-p — nom1 — nom2 ...

7.2.5 ـ الضرب : MULTIPLY

الشكل 1 :

MULTIPLY nom 1 BY nom 2 [ROUNDED]

nom1 هو عبارة عن إسم متحولة أو ثابتة .

nom2 هـو عبارة عن إسم المتحـولة التي ستحتـوي على نتيجـة ضرب nom1 بـ nom2 .

الشكل 2 :

MULTIPLY nom 1 BY nom 2 GIVING nom 3 [ROUNDED]

nom1 وnom2 هي عبارة عن ثوابت أو متحولات .

nom3 هي عبارة عن المتحولة التي ستحتوي على نتيجة ضرب nom1 بـ nom2 .

7.2.6 ـ القسمة : DIVIDE

الشكل 1 :

DIVIDE nom 1 INTO nom 2 [ROUNDED]

nom1 هي عبارة عن إسم متحولة أو ثابتة .

nom2 هي عبارة عن المتحولـة ، والتي ستحتوي عـلى نتيجة قسمـة nom2 على nom1 .

الشكل 2 :

DIVIDE nom 1 INTO nom 2 GIVING nom 3 [ROUNDED]

nom1 وnom2 هي عبارة عن متحولات أو ثوابت .

nom3 هي عبارة عن المتحولة التي ستحتوي على نتيجة قسمة nom2 على nom1 .

7.2.7 ـ التعليمة العامة COMPUTE

هذه التعليمة تسمح بكتابـة تعابـير في لغة كـوبول شبيهـة بتلك المكتوبـة في لغة

فورتران . وهي تسمح إذاً باستبدال الأربعة أفعال السابقة .

الشكل :

COMPUTE nom 1 = expression.

nom1 هي متحولة .

Expression هي عبارة عن ثابتة ، متحولة أو تعبير جبري مختلف . كتابة التعابير هو نفسه كما في فورتران ، مع إستعمال الأهلّة البسيطة ، ونفس قواعد أفضلية المؤثرات (بالنسبة لـ + ، - ، × ، ÷) .

لنشر إذاً إلى أن النحو في لغة كوبول يتطلب أن تكون كل سمة مؤثر (+، -، *، /) مسبوقة ومتبوعة بفراغ .

7.3 ـ التعابير المنطقية

7.3.1 ـ مؤثرات المقارنة

هناك ثلاثة مؤثرات للمقارنة في لغة كوبول ، تُمثّل هذه المؤثرات إما بالكتابة حرفياً ، وإما بواسطة إشارة خاصة . وهي :

GREATER THAN	>	أكبر من
EQUAL TO	=	يعادل
LESS THAN	<	أصغر من

الكلمات «THAN» و«TO» هي إختيارية .

مثلاً :

A GREATER THAN B

هو تعبير منطقي بقيمة « صحيح » إذا كان A أكبر من B ، و« غلط » في الحالة المعاكسة ، أي إذا كان A أصغر أو يعادل B .

القواعد :

أ ـ عمليات المقارنة تطبّق على الثوابت الحرفية (Literal) ، أو على المتحولات ، أو أيضاً على التعابير الجبرية .

ب ـ إذا استعملنا السمات < = > ، فمن الأنسب إتباعها أو إستباقها ببياض واحد على الأقل .

7.3.2 ـ مؤثرات منطقية أساسية

هي عبارة عن المؤشرات الكلاسيكية الثلاثة، التي نجدها في جميع لغات البرمجة .

NOT	إعكاس
OR	إتحاد
AND	تقاطع

وإذا كانت A وB عبارة عن متحولات (أو تعابير) منطقية ، فالقواعد التالية تنطبق :

NOT A
- حقيقة إذا كان A غلط
- « غلط» إذا كان A «حقيقة»

A OR B
- حقيقة إذا كان A وB حقيقة
- حقيقة إذا كان A حقيقة و B «غلط»
- حقيقة، إذا كان A غلط وB « حقيقة»
- غلط إذا كان A وB غلط

A AND B
- حقيقة إذا كان A وB حقيقة
- غلط إذا كان A حقيقة وB حقيقة
- غلط إذا كان A غلط وB حقيقة
- غلط إذا كان A وB غلط

القواعد :

من الممكن كتابة تعابير منطقية كبيرة وطويلة كما نرغب ، وبنوعين من المؤثرات . الأوليات التالية تنطبق عليها وهي :

(1) المؤثرات الجبرية .
(2) مؤثرات المقارنة
(3) المؤثر NOT
(4) المؤثر AND
(5) المؤثر OR .

من الممكن أيضاً تعديل قاعدة الأولوية باستعمال الأهلّة .

7.4 ـ مؤثرات المراقبة والتحكّم

7.4.1 ـ انقطاع غير مشروط

GO TO simple

الشكل :

```
GO TO nom
```

nom هو إسم الفقرة أو القسم وقد يكون وسماً.

ب ـ الأمر GOTO غير المستقل

الشكل :

```
GO TO nom 1 [nom 2] ... DEPENDING ON nom 3
```

nom 1, nom 2, ... هي عبارة عن أسماء فقرات أو أقسام .

nom 3 هي متحولة تحتوي على قيمة صحيحة موجودة بين 1 و n ، n هي عدد أسماء الفقرات أو الأقسام الموجودة خلف الأمر . GO TO

ج ـ ALTER

هذا الأمر يسمح بتعديل تشغيل التعليمات GO TO ، وذلك بتغيير العناوين ديناميكياً .

الشكل :

```
ALTER nom 1 TO PROCEED TO nom 2.
```

القواعد :

1 ـ إذا كان الاسم nom 1 وهو وسم GOTO غير متبوع بواسطة وسم ، فالأمر GO TO هو كامل .

مثلاً :

```
PARAG1.   GO TO.
.
.
ALTER PARAG1 TO PROCEED TO SUITE1.
.
.
```

عندما تتم تنفيذ الأمر GO TO ، فكل شيء يسير كما لو كان البرنامج يحتوي على التعليمة :

```
PARAG1.   GO TO SUITE1.
```

2 ـ عندما يتم تكملة الأمر GO TO ، فالتعليمة الجديدة ALTER تسمح بتعديل التفريع .

مثلاً :

```
ALTER PARAG1 TO PROCEED TO SUITE2.
```

الفقرة موضع السؤال تعمل الآن كما يلي :

```
PARAG1.  GO TO SUITE2.
```

7.4.2 ـ الانقطاع المشروط

الشكل :

```
IF expression THEN instruction 1
              [ELSE instruction 2]
```

expression هو عبارة عن تعبير منطقي .

Instruction 2 ، Instruction 1 هي عبارة عن تعليمات كوبول ، قابلة للاستبدال محتملاً بواسطة NEXT SENTENCE .

قاعدة :

1 ـ الكلمة THEN هي إختيارية .

2 ـ القسم الثاني من التعليمة ، الذي يبدأ بواسطة ELSE ، هو إختياري . ولكن إذا جرى إستعماله ، فالكلمة ELSE هي إلزامية .

التشغيل : إذا كان التعبير المنطقي هو حقيقي ، فسيتم تنفيذ التعليمة instruction1 . وإلا ، فسيتم القفز عن التعليمة Instruction 1 وسيجري تنفيذ التعليمة Instruction2 . وفي غياب Instruction2 ، يُتابع البرنامج على التوالي .

القاعدة :

من الممكن تداخل IF ، أي إستعمال IF في مكان التعليمة Instruction 1 أو Instruction 2 . هذه الطريقة لا يُنصح بها غالباً ، لاسباب ناتجة عن تعقيد البرنامج منطقياً ، والذي قد يُسبِّب أخطاءً .

7.5 ـ الحلقات

الحلقات ، في لغة كوبول ، تتم بواسطة التعليمة PERFORM . هذه الأخيرة تستعمل بأشكال كثيرة .

أ ـ الشكل 1 :

```
PERFORM nom 1 [THRU nom 2]
```

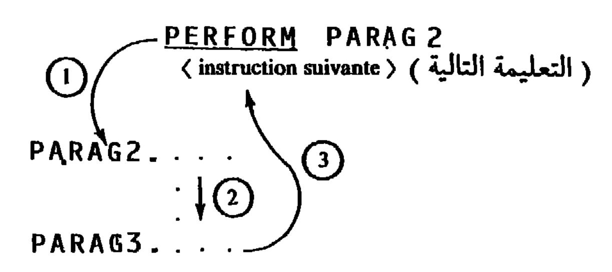
nom 1 و nom 2 هي عبارة عن أسماء فقرات أو أقسام .

التشغيل : التعليمة PERFORM تنقل التحكّم إلى بداية الفقرة (أو القسم) nom 1 . في نهاية الفقرة (أو القسم) nom 1 (أو في نهاية الفقرة nom 2 عند الاقتضاء) ، يُعاد التحكم إلى التعليمة التي تتبع PERFORM .

القواعد :

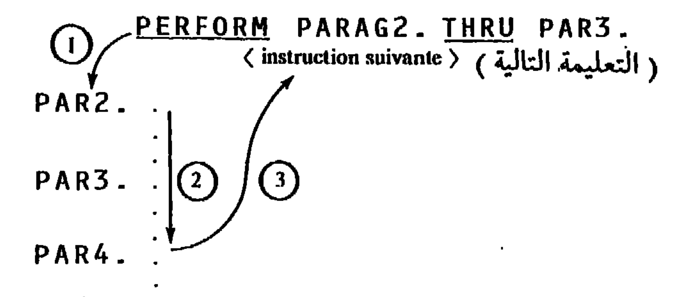
كي يتم تنفيذ التعليمة PERFORM بشكلٍ صحيح ، يجب أن يتم بلوغ نهاية nom 1 (أو nom 2) بشكلٍ فعلي . وإذا كان هناك تفريعات في nom 1 ، فيجب أن تُعاد (تُرجع) إلى nom 2 ، الذي لا يحتوي إلا على التعليمة الوحيدة EXIT .

مثلاً 1 :



(بداية الفقرة تدل على نهاية السابقة) .

مثل 2 :



الشكل 2 : .

PERFORM nom 1[THRU nom 2] nom 3 TIMES

nom 1 وnom 2 لها نفس التعريف أعلاه .

nom 3 هي عبارة عن متحولة صحيحة أو ثابتة صحيحة إيجابية .

التشغيل : تُنفـذ التعليمة PERFORM كما في شكلها البسيط (الذي رأيناه سابقاً في أ) ، ولكن لعدد من المرّات مُحدَّد بواسطة nom 3 .

ج - الشكل 3 :

PERFORM nom 1[THRU nom 2] UNTIL expression

expression هـو تعبير منـطقي . طالما هذه التعليمـة هي بقيمة « غلط False »

فسيجري تنفيذ التعليمة PERFORM . وإذا كان التعبير expression هو « حقيقة TRUE » ، فستهمل التعليمة PERFORM والتحكّم في البرنامج سيتابع على التوالي .

د ـ الشكل العام

```
PERFORM nom 1 THRU nom 2 VARYING nom 3
 FROM nom 4 BY nom 5 UNTIL expression AFTER...
```

nom 1 وnom 3 تُعرَّف كما في الفقرة (أ) .

nom 3 هي عبارة عن متحولة .

nom 4 وnom 5 هي ثوابت أو متحولات .

expression هو تعبير منطقي .

من الممكن تكرار المتتالية UNTIL ... VARYING بعد AFTER

التشغيل : تدور أو تُنفَّذ التعليمة PERFORM لجميع القيم من nom 3 حتى nom 4 بخطوة تعادل nom 5 وحتى يأخذ التعبير المنطقي القيمة « حقيقة TRUE » .

أمثلة :

```
PERFORM PAR4 VARYING A FROM 10 BY 2
UNTIL A = 20.
```

سيتم تنفيذ الفقرة PAR 4 للقيم المتتالية A من 10, 12, 14, 16 و18 (ولكن بدون القيمة 20) .

هـ ـ تداخل PERFORM

من الممكن تداخل الأوامر PERFORM : وهذا يعني إن PERFORM سيتم إدخالها في فقرة (أو في قسم) تُستدعى بواسطة تعليمة PERFORM أخرى ، وهكذا دواليك . يجب الاعتراف في هذه المرّة أن الكوبول هو أصعب من فورتران PL/ 1 في هذا النوع من المسائل ، وفي كل ما يتعلّق بالبرنامج .

7.6 ـ إدخال ـ إخراج بسيط

7.6.1 ـ في القراءة : ACCEPT

الشكل :

```
ACCEPT nom[FROM CONSOLE]
```

في غياب FROM CONSOLE ، يقرأ هذا الأمر البطاقات . يجري التحكم بعدد السمات بواسطة nom . وإذا جرى إستعمال FROM CONSOLE ، فالإدخال يتم من

خلال لوحة الملامس التابعة لقنصلة التحكّم في الحاسب . وهنا أيضاً يجري التحكم بعدد السمات الداخلة بواسطة nom .

7.6.2 ـ في الكتابة : DISPLAY

الشكل :

DISPLAY nom 1[nom 2] ... [UPON CONSOLE]

nom 1, nom 2 , ... هي عبارة عن متحولات أو ثوابت .

وإذا جرى إستعمال UPON CONSOLE ، فسيتم إرسال سلسلة السمات إلى منضدة التحكم بالمكنة (pupitre) (أي إلى الشاشة مثلاً) . وإلا ، إلى الطابعة .

7.7 ـ إدخال ـ إخراج طبيعي

جميع عمليات الإدخال والإخراج الطبيعية تذهب مزدوجة مع وصف السجلات في (ENVIRONNEMENT DIVISION و DATA DIVISION) سيتم فتح كل سجل ، وسنقوم بعمليات إدخال ـ وإخراج فيه . وفي نهاية العمل ، سيتم إغلاقه ، كي يتم لاحقاً إعادة فتحه عند الحاجة .

7.7.1 ـ أمر الفتح : OPEN

الشكل :

OPEN {INPUT | OUTPUT | I-O} nom 1 nom 2 ...

nom 1 ، nom 2 هي عبارة عن أسماء سجلات ، موجودة خلف عدد FD في قسم المعطيات DATA DIVISION .

INPUT وتعني فتح السجل عند القراءة .

OUPTUT تعني إغلاق السجل عند الكتابة .

I-O تعني فتح للقراءة والكتابة (عشوائي RANDOM فقط) .

الصيغ الممكنة :

1 ـ WITH NO REWIND لا يوجد إعادة لف للسجل عند الفتح . تستعمل هذه الصيغة عند إستعمال أشرطة متعددة السجلات .

2 ـ REVERSED . تستعمل في القراءة فقط . يُقرأ السجل في الاتجاه المعاكس .

7.7.2 ـ أمر القراءة READ :

الشكل 1 (متتالي) .

READ nom 1[INTO nom 2] AT END instructions.

nom 1 هو إسم السجل .

nom 2 هو إسم متحولة . نسخة عن التسجيلة يتم إرسالها حسب قواعد التعليمة MOVE ، في nom 2 ، إذا كانت الصيغة INTO هي المستعملة . . .

الشكل 2 (بلوغ مباشر) .

READ nom 1[INTO nom 2] INVALID KEY instruction.

في جميع الحالات ، سيتم إنتقال مضمون تسجيلة في كل مرة إلى المعطى المرتبط بالسجل (DATA RECORD IS ...) .

7.7.3 ـ أمر الكتابة : WRITE

الشكل 1 (متتالي) .

WRITE nom 1[FROM nom 2] AFTER ADVANCING nom 3

nom 1 ، هو إسم التسجيلة .

nom 2 هو إسم متحولة حيث المعلومات هي منسوخة قبل أي شيء إذا كانت الصيغة FROM مستعملة .

nom 3 هو عبارة عن ثابتة إيجابية ، أو متحولة أو كود حرفي ، يتحكم بالطفور عن الأسطر . وإذا كان في عبارة عن كود حرفي ، فيجب مراجعة سمة قد تكون إحدى السمات التالية :

طفور بسيط عن سطر	(بياض)
طفور عن صفحة	1
طفور عن سطرين	0
لا يوجد طفور	+

الصيغة AFTER (طفر عن سطر قبل القراءة) يُمكن أن تُستبدل بواسطة BEFORE (طفور عن سطر بعد القراءة) .

الشكل المُبسَّط : في الحالة التي لا يجب فيها طباعة السجل، سيكون شكل التعليمة أبسط :

WRITE nom 1[FROM nom 2]

شكل 2 (بلوغ مباشر) .

```
WRITE nom 1[ FROM nom 2] INVALID KEY instruction
```

7.7.4 ـ أمر الإغلاق : CLOSE

الشكل :

```
CLOSE nom 1[clause] nom 2[clause] ...
```

nom 1, nom 2, ... هي عبارة عن أسماء سجلات .

البند (clause) يُمكن أن يكون :

1 ـ WITH NO REWIND الـذي يعني أن السجل لن يتم إعـادة لفّه بعـد الإغـلاق (الإنشاء على أشرطة متعددة سجلات) .

2 ـ WITH LOCK ، الذي يعني إن السجل لا يُمكن أبداً أن يُستعمل بواسطة البرنامج الجاري .

8 ـ الفرز بلغة كوبول

عمليات فرز السجلات هي أساسية في الإدارة . من الضروري أن يكون النظام المستعمل مجهزاً ببرنامج فرز أكثر فعالية قدر الإمكان .

وتعتمد لغة الكوبول تعليمة خاصة للفرز ، سهلة الإستعمال ، وفائدتها بديهية .

8.1 ـ وهو سجل الفرز في القسم DATA DIVISION

في عملية الفرز ، هناك سجل دخل مطلوب فرزه ، وسجل إخراج هو السجـل المفروز ، إضافة إلى بعض التعديلات ، وإلى سجل « عمل » .

الشكل :

```
SD nom RECORDING MODE ....
RECORD CONTAINS ....
DATA RECORD ....
```

سجلات الإدخال والإخراج هي سجلات متتالية كلاسيكية .

8.2 ـ تعليمة الفرز SORT

توجد هذه التعليمة ، طبيعياً ، في القسم PROCEDURE . السجل هو مفتوح أوتوماتيكياً بواسطة النظام .

الشكل الأساسي :

```
SORT nom-fichier-tri ON { ASCENDING  } KEY nom 1[nom 2]...
                        { DESCENDING }
     { USING                nom-fichier-entrée }
     { INPUT  PROCEDURE     nom-p-1            }
     { GIVING               nom-fichier-sortie }
     { OUTPUT  PROCEDURE    nom-p-2            }
```

nom-fichier-tri الموصف بواسطة SDS في القسم DATA DIVISION

nom 1, nom 2, ... هي عبارة عن معطيات التسجيلة الموجودة في سجل الفرز .

nom-p-1, nom p-2 هي أسماء الأقسام ، وتُسجَّل إلزامياً .

INPUT PROCEDURE هدف هذا القسم هو إنشاء سجل الفرز . ويحتوي إلزامياً على أمر بالكتابة .

جميع السجلات المستعملة في INPUT PROCEDURE هي بتشغيل كلاسيكي .

OUTPUT PROCEDURE هذا القسم يجمع عناصر سجل الفرز . ويحتوي إلزامياً على أمر بالقراءة :

```
RETURN nom-fichier-tri [INTO nom donnée]
AT END instructions.
```

9 ـ البرامج الثانوية

من الممكن عادة إستعمال برامج ثانوية في الكوبول . هذا الاتجاه هو غير معيَّر ، وينبع من إتجاهات مختلفة حسب المُصمِّم .

لهذا السبب ، ولهذا القسم ، ينصح القارىء بمراجعة الملاحظات الخاصة بالنظام المستعمل .

ملحق : كلمات محجوزة للغة

لا يمكن إستعمال الكلمات التالية كإسم للمعطيات ، أسماء إجراءات أو أسماء - شروط .

A
ACCEPT
ACCESS
ACTUAL
ADD
ADVANCING
AFTER
ALL
ALTER
AND
APPLY
ASCENDING
ASSIGN
AT
AUTHOR
B
BASIS
BEFORE
BLANK
BLOCK
BY
C
CALL
CHARACTERS
CLOSE
CODE
COLUMN
COMP
COMPUTATIONAL
COMPUTE
CONSOLE
CONTROL
COPY
CORE-INDEX
CORR
CORRESPONDING
CSP
OMITTED
ON
OPEN
OPTIONAL
OR
OTHERWISE
OUTPUT
P
PAGE
PAGE-COUNTER
PERFORM

CURRENCY-SIGN
CURRENT-DATE
C Ø1 — C Ø2 — C12
D
DATA
DATE
DECIMAL
DELETE
DEPENDING
DESCENDING
DETAIL
DISP
DISPLAY
DIVIDE
DIVISION
DOWN
DUMMY
E
EJECT
ELSE
END
ENTER
ENTRY
EOF
EXAMINE
EXIT
F
FD
FILE
FILLER
FINAL
FOR
FREE
FROM
GENERATE
GIVING
GO
REMARKS
RENAMES
REPLACING
REPORT
REREAD
RERUN
RESERVE
RESET
RETURN
REVERSED
REWRITE

GOBACK
GROUP
HIGH
ID
IF
IN
INDEX
INDEXED
INITIATE
INPUT
INSERT
INSTALLATION
INTO
I-O
I-O-CONTROL
INVALID
JUSTIFIED
KEY
LABEL
LEADING
LEAVE
LINE
LINE-COUNTER
LOW
MEMORY
MOVE
MULTIPLE
MULTIPLY
NEXT
NO
NOMINAL
NOT
NOTE
NULLFILE
OBJECT-COMPUTER
OCCURS
OF
SUM
SUPPRESS
SYNCHRONIZED
TALLY
TERMINATE
THEN
THRU
TIMES
TO
TRACE
TRANSFORM

PIC	ROUNDED	TYPE
PICTURE	RUN	U
POSITION	S	UNIT
POSITIONING	SAME	UNTIL
PRINT-SWITCH	SEARCH	UP
PROCEDURE	SECURITY	UPDATING
PROCESSING	SEEK	UPON
PROGRAM-ID	SEGMENT-LIMIT	USAGE
QUOTE	SELECT	USE
QUOTES	SET	USING
RD	SIZE	UTL
READ	SORT	V
READY	SOURCE	VALUE
RECORD	SPACE	VARYING
RECORDING	SPACES	WHEN
RECORDS	SPECIAL	WITH
REDEFINES	STANDARD	WRITE
REEL	START	ZERO
RELEASE	STOP	ZEROES
REMAINDER	SUBTRACT	ZEROS

مراجع :

مراجع أساسية

الوثائق التقنية التابعة للصانعين ، والخاصة بلغة كوبول والتحديدات الداخلية الخاصة .

من الممكن أيضاً إستشارة :

J VIGNES, C. COURTY-LAJEUNESSE et C. DEBOST-BENTZ, *Théorie et pratique de la programmation COBOL*, Editions TECHNIP, (Paris, 1972).



الفصـل الثـامن

لغة PL / 1

1 ـ عموميات

1.1 ـ مجموعة السمات

يوجد عدة مجموعات سمات في لغة PL/ 1 . والمجموعة الوحيدة المستعملة عملياً تتألف من 60 سمة . أما المجموعة من 48 سمة فهي تستدعي حجز بعض كلمات المفاتيح وهي ليست مستعملة بشكل عملي ، ولن يتم مناقشتها هنا .

يوجد ثلاثة أصناف من السمات :

ـ السمات الأبجدية .

ـ السمات الرقمية (الأرقام) .

ـ السمات الخاصة .

أ ـ السمات الأبجدية . عدد : 29

وهي عبارة عن 26 سمة من الألفباء ، من A إلى Z ، والسمات الثلاث التالية :

ـ $ سمة العملة .

ـ @ السمة «at» أو «arrobas»

ـ # السمة التي تدعى « رقم » .

ب ـ السمات الرقمية ، وعددها 10

هي عبارة عن الأرقام العربية من 0 إلى 9 (*)

ج ـ السمات الخاصة وعددها 21

السمة الأولى هي البياض (الفراغ)

هناك بعد ذلك البياض بخط أفقي : ـ (ويدعى أيضاً Break) .

السلسلة التالية ، وتتألف على صعيد المثال ، من السمات B, C ، ومن السمات D, E, F .

ABC_DEF

(*) يُكتب الصفر هنا بدون خط عامودي في وسطه (0) .

هناك بعد ذلك السمات التالية :

= يعادل

+ زائد

– ناقص

/ قضيب القسمة «Slash»

(هلال يسار أو مفتوح

) هلال يمين أو إغلاق

. نقطة

، فاصلة

؛ نقطة فاصلة

: نقطتين

% بالمئة

¬ معكاس (مؤثر منطقي)

& و(« و » مانعة) .

| أو (« أو » مشتملة) .

> أكبر من

< أصغر من

؟ علامة إستفهام .

* نجمة .

1.2 ـ المعرِّفات (Identificateurs)

تتألف المعرِّفات من السمات الأبجدية والسمـات الرقمية والخط الأفقي (ـ) . وتظهر في موضع آخر خارج الملاحظيات . وتكون مسبوقة ومتبوعة بواسطة فاصل . السمة الأولى من المعرِّف تكون دائماً أبجدية . وفي النهاية ، الطول الأقصى هو 31 سمة . جميع السمات الخاصة هي فواصل ، ما عدا الخط الأفقي (ـ) .

وتلعب السمة « بياض (فراغ) » دوراً أساسياً في PL/ 1 . وعندما لا يكون هناك حاجة إلى أية سمة خاصة ، تُفصل المعرِّفات إلزامياً بواسطة فراغ واحد على الأقل . ما عدا في سلسلة من السمات ، أو في سلسلة مُؤلفة من عدد معيَّـن من الفراغات (البياضات) بقيمة فراغ واحد . وفي النهاية يُمكن للفراغ أن يُوضع في أي مكـان بداخـل المعرِّفـات والفواصل .

مثلاً :

هي تعليمة صحيحة `GO TO SUITE;`

تعليمة بدون معنى

```
GO TOSUITE;
```

هي تعليمة بدون معنى لأنه يجب أن تُفصل الكلمات 'TO' و«SUITE» عن بعضها بواسطة فراغ واحد على الأقل .

وفي لغة PL / 1 يُمثِّل المعرَّف :

- متحولة لا إتجاهية
- جدول
- تركيبة
- أقسام ثانوية من التركيبة .
- وسم
- نقطة دخول
- سجل
- كلمة مفتاح (أمر) .

هناك إختصارات لبعض الكلمات ـ مفاتيح

الأمثلة التالية هي عبارة عن مُعرَّفات :

```
A
FIN_DE_LECTURE
WRITE
DECLARE
```

وعلى العكس ، فالأمثلة التالية ليست معرَّفات :

تبدأ بواسطة رقم 4526_TR

تحتوي على سمة خاصة (-) ALPHA-X

وفي النهاية ، لا يوجد كلمات محجوزة في لغة PL / 1 .

1.3 ـ المعطيات في لغة PL / 1

تتمتع لغة PL/ 1 بعدد كبير من المعطيات ، وهذا هو مصدر قوتها .

ترتيب	نوع المعطيات	الخاصية
معطيات المسألة :	ـ جبرية	BINARY, DECIMAL, FIXED, FLOAT, REAL, COMPLEX, PICTURE
	ـ سلاسل	CHARACTER, BIT, VARYING, PICTURE

معطيات تحكّم بالبرنامج	ـ أوسمة (LABELS)	LABEL
	ـ أعمال (مبرمجة متعددة)	TASK
	ـ إلتقاء (برمجة متعددة)	EVENT
	ـ مؤشرات	POINTER, OFFSET
	ـ حيزات	AREA
أسماء مداخل	أسماء مداخل	ENTRY, RETURNS
سجلات	سجلات	FILE

1.4 ـ التعليمة DECLARE

يُصرّح عن جميع المعطيات قبل إستعمالها ، بواسطة تعليمة تُضيف معرّفاً مع عدد من الخاصيات إليها . هذه الخاصيات تصف بشكلٍ دقيق المعطيات المنشأة . من الممكن أحياناً إحتمال واحد أو عدة تصريحات . ويقوم المُصرِّف بنفسه بالتصريح لها حسب مفهومها . هكذا مثلاً وبشكل عام لا تُصرّح أبداً بصراحة عن الأوسمة .

الشكل هو :

<u>DECLARE</u> *identificateur* attribut-1 attribut-2 ...;

مثلاً :

```
DECLARE ALPHA DECIMAL FIXED;
DECLARE TEXTE CHARACTER(2000) VARYING
INITIAL(' ');
```

أو ، بإيجاز :

```
DCL ALPHA DEC FIXED;
DCL TEXTE CHAR(2000) VAR INIT(' ');
```

إضافة لذلك ، يُمكن أن تُجمّع التصريحات في نفس الأمر DECLARE :

```
DECLARE:
DCL ALPHA DEC FIXED, TEXTE CHAR(2000)
VAR INIT(' ');
```

القاعدة :

لا يُمكن التصريح عن معرّف معيّن وبوضوح في نفس الأمر . في غياب خاصية أو عدة خاصيات لمعطى معين ، تأخذ PL / 1 الصيغ بالغلط (par defaut) .

1.5 ـ نسق الكتابة والتركيبة العامة للبرنامج

1.5.1 ـ نسق الكتابة

تكتب التعليمات في لغة PL / 1 في نسق حرّ .

وإذا استعملنا البطاقات كناقل للإدخال، فيمكن أن يجري تثقيب التعليمات من العامود 1 إلى العامود 80 . وعملياً ، لا يُعالج المصرِّف سوى الأعمدة من 2 إلى 72 . ولكن هذه الأعمدة هي إختيارية ، ويُمكن أن يجري تعديلها بواسطة تعليمات خاصة بالمصرّف .

تنتهي التعليمة بواسطة نقطة ـ فاصلة . وهذه هي النقطة ـ الفاصلة التي توقف التعليمة ، وليس نهاية البطاقة . التعليمة صفر هي النقطة ـ الفاصلة . ومن الممكن أن نضع عدة تعليمات على نفس السطر :

```
A = 0; BETA=A*B/TA(54,7) + SQRT(X); PUT
DATA;
```

وفي النهاية ، من الممكن أن نضع عدداً من الفراغات غير محدد بين مختلف المعرِّفات .

1.5.2 ـ الملاحظية (Commentaires)

من الممكن أن نكتب الملاحظية في أي موقع من البرنامج يُمكن أن يظهر فيه الفراغ ، ما عدا في الثابتة من نوع CHARACTER . الملاحظية هي سلسلة من السمات المختلفة . وتبدأ بسلسلة من سمتين : * / .

وتنتهي أيضاً بنفس هذه السلسلة من السمات : / * . السمات / و * يجب أن تتابع مباشرة ، بدون أي فراغ فيما بينها .

يُمكن للملاحظية أن تكون بأي طول ممكن . تبدأ الملاحظية أوتوماتيكياً عندما يجد المصرِّف السلسلة * / وتنتهي عندما يلتقي أول سلسلة / *..

أمثلة :

```
/* CECI EST UN COMMENTAIRE */
```

1.5.3 ـ مجموعات وقدرات من البرنامج

المجموعات (أو مجموعات DO) هي عبارة عن سلسلة من التعليمات الموجودة بين DO والتعليمة END .

مثلاً :

```
ALPHA: DO;
       A=0; B,C,X = 1;
       PUT FILE (EDITEUR) LIST (K1,K2);
       CALL PSC;
       END;
```

من الممكن أيضاً أن نكتب هذه المجموعة حسب الطريقة التالية :

```
ALPHA: DO; A=0; B,C,X = 1; PUT FILE
(EDITEUR) LIST(K1,K2); CALL PSC; END;
```

التعليمات هي متتالية الواحدة بعد الأخرى دون أخذ بالحسبان الفواصل الفيزيائية للبطاقات .

الفدرات (Blocs) هي أيضاً عبارة عن سلاسل من التعليمات ، ولكنها تخضع لبعض القواعد ، وبالتحديد إلى كل ما يتعلّق بالبرنامج وبالتعرّف على الأسماء .

وهي نوعان: الفدرات BEGIN والفدرات PROCEDURE . وتركيبتها هي التالية :

```
K: BEGIN; ... ; ... ; ... ; ... ; END
K;
PA: PROCEDURE; ... ; ... ; ... ; ... ;
END PA;
```

يجب أن تكون الفدرة BEGIN متداخلة في إجراء من نوع PROCEDURE ويجري تنفيذها على التوالي .

يجب أن يتم مناداة الأجراء PROCEDURE بواسطة CALL .

يجب أن تكون جميع الفدرات والمجموعات التي تؤلف البرنامج متداخلة في إجراء رئيسي (procedure principale) ، يصبح فعالاً بشكل أوتوماتيكي بواسطة النظام .
ولكن من الممكن أن يكون هناك إجراءات خارجية (External procedure) يُمكن أن يتم طلبها بواسطة الأجراء الرئيسي . وللأجراء الرئيسي الخاصية OPTIONS (MAIN) .

```
PP: PROCEDURE OPTIONS(MAIN);
  < instruction > ;
  < instruction > ;
  END PP;
```

سيبدأ البرنامج PL / 1 الكامل ودائماً بواسطة التعليمة

```
PROCEDURE OPTIONS(MAIN);
```

أو باختصار

```
PROC OPTIONS(MAIN);
```

وينتهي فيزيائياً بواسطة التعليمة END المناسبة للتعليمة

```
PROCEDURE OPTIONS (MAIN);
```

تخصيص الذاكرة يُمكن أن يكون ساكناً (يتم قبل تنفيذ البرنامج) أو ديناميكياً (خلال التنفيذ).

بشكل ساكن ، يبقى تخصيص الذاكرة صالحاً خلال كامل مدة البرنامج ، بينما ديناميكياً هو ليس كذلك ، أي إن تخصيص الذاكرة يتم لفدرة معينة وليس أثناء عمل الفدرة . هناك أربعة طبقات لتخصيص الذاكرة في لغة PL / 1 .

- ساكن (STATIC)
- أوتوماتيكي (AUTOMATIC)
- مُراقب (CONTROLLED)
- مُباشر (BASED)

الطبقة المختارة ضمنياً هي AUTOMATIC ما عدا إذا جرى التصريح عن المتحولة EXTERNAL (بالغلط هي إذاً STATIC) .

1 ـ ساكن

مثلاً :

```
DECLARE X FIXED DECIMAL(10,5)
STATIC;
```

الإدارة هي إذاً شبيهة بكل ما يجري في لغة فورتران . تُخصَّص المواقع في الذاكرة مرة واحدة والباقي خلال مدة دوران البرنامج .

نصرِّح إذاً عن متحولة ساكنة إذا كنا نرغب في أن تكون معروفة من جميع الفدرات التي تؤلِّف البرنامج وإذا كانت تستخدم في كل مكان ، نربح وقتاً إذاً عند التنفيذ .

2 ـ أوتوماتيك (AUTOMATIC)

بشكل عام لا نقوم بالتصريح ، لأن هذه الصيغة هي مأخوذة بالغلط (ما عدا للمتحولات من نوع EXTERNAL) .

تخصيص الذاكرة ، بجميع متحولات الفدرة ، يتم عندما تصبح الفدرة فعالة . وكل ذلك يتم خلال مدة التحكم المدعوة «Prologue» .

مثلاً :

```
PA: PROCEDURE OPTIONS (MAIN);
    (X,Y,Z) مجهولة ، لا يوجد حيز محجوز من الذاكرة

    CALL PSC; ——————→ تخصيص مكان في
                       الذاكرة للمتحولات Z, Y, X .
    ←——————————┐       تنفيذ PSC ← END
    .          │       épilogue : تحرير الموقع المشغول
    .          │       في الذاكرة من قِبل Z, Y, X
    .          └——————
PSC_PROCEDURE;
    DCL(X,Y,Z) FIXED;
    .
    .
    END PSC;
    .
    .
    END PA;
```

3 ـ مُراقب

في هـذه الحـالـة ، يتم التخصيص حسب الأمـر ، وبـواسـطة تنفيـذ التعليمـة ALLOCATE . التخصيص يبقى ، ولكن بعد الإنتهاء من الفدرة . الأمر FREE يُحرِّر الذاكرة المشغولة سابقاً .

التعليمات المتتالية ALLOCATE تكدس حيزات الذاكرة . وإذا أردنا استعمال وتغـطية حيـزات مشغولـة سابقـاً ، علينا أن نستعمـل الأمر FREE قبـل الأمر الثـاني ALLOCATE .

4 ـ المباشر أو المؤشر

مثلاً :

```
DECLARE Z BASED(P);
```

حيث Z هي متحولة وP هي مؤشر (مؤشر مطلق) .

في الحالة الأخيرة نراقب مباشرة موقع المتحـولات في الذاكـرة ، بواسـطة عنوانها الحقيقي .

التشغيل هو شبيه بالحالة CONTROLLED ، مع فارق بسيط وهو في كون جميع الحيزات مبلوغة وليس فقط تلك الموجودة في « أعلى » المكدس .

تُستعمل الإدارة والتنظيم المباشر للذاكرة عند معالجة اللوائح مـع المؤشر المُطلق

(POINTER) والنسبي (OFFSET) . وفي النهـايـة ، وبسبب الإدارة الـديناميكيـة للذاكرة ، فإن التكـرار هو ممكن في لغـة PL/ 1 . ولكن عرضـه يخرج عن إطـار هذا الكتاب ، ويُنصح المستعمل بالعودة إلى المراجع .

1.6 ـ معطيات اللغة

1.6.3 ـ المعطيات الجبرية

وتحدد بواسطة قاعدتها ، معيارها ، وقتها وطريقة تمثيلها .

القاعدة : عشـري أو ثنائي .

المعيار : فاصلة ثابتة أو فاصلة متحركة .

الدقة : عدد الأرقام ذات الدلالة .

الطريقة : حقيقي أو مركب .

أ ـ المعطيات الحقيقية

1 ـ بفاصلة ثابتة

وغالباً هي الأكثر إستعمالاً . وتتألف من قسم صحيح و ، إختيـارياً ، من قسم عشري .

أمثلة :

```
72.192
  .5
4536
   0
```

وتتألف من 15 رقماً ذوو دلالة كحد أقصى . الدقة بالغلط ، أي بدون تحديد خاص هي (5,0) . وهذا يعني 5 أرقام بالمجموع و0 بعد الفاصلة العشرية .

أمثلة على التصريح عن المتحولات العشرية بفاصلة ثابتة :

```
DECLARE A DECIMAL FIXED(8,2), B DEC
FIXED(14);
```

تتألف A من 8 أرقام ذات دلالة ، منها اثنين بعد الفاصلة العشرية . تتألف B من 14 رقماً بدلالة ، حيث 0 بعد الفاصلة العشرية .

2 ـ ثنائي بفاصلة ثابتة

وتحتوي على واحد أو عدة أرقام ثنائية ، وبشكل إختياري ، على نقطة تفصل القسم الصحيح من القسم الكسري ، ومتبوعة بالحرف B وبدون أي فراغ بين الأرقام .

أمثلة :

10110B	(يعادل 22)
101.01B	(يعادل 5.25)
.101B	(يعادل 625.)

وتحتوي هذه الأعداد على 31 رقماً ثنائياً . والدقة بالغلط هي (15,0) .

مثل على التصريحات عن المتحولات الثنائية بفاصلة ثابتة :

```
DECLARE KB BINARY FIXED(18), F BIN
FIXED(19,6);
```

تتألف المتحولة KB من 18 رقماً ثنائياً بدلالة ، حيث 0 بعد الفاصلة .
أما المتحولة F فتتألف من 19 رقماً بدلالة ، حيث منها 6 بعد الفاصلة العشرية .

3 ـ العشرية بفاصلة متحركة

يدخل هنا أس يتألف من رقمين ، إيجابي أو سلبي ، هو عبارة عن القوة من 10 .

أمثلة :

```
12E+23
317.5E-16
0.1E3
.42E+50
```

الدقة القصوى هي 16 رقماً . الدقة الضمنية بالغلط هي من 6 أرقام . أمثلة على التصريح عن المتحولات بفاصلة متحركة :

```
DECLARE  CCR DECIMAL FLOAT(16),
         XA4 FLOAT DEC (10),
         YFHA6A8 DEC FLOAT;
```

CCR بدقة من 16 رقماً عشرياً .
XA4 بدقة من 10 أرقام .
YFHA6A8 بدقة من 6 أرقام بالغلط .

4 ـ الأعداد الثنائية بفاصلة متحركة

وتتألف من أس من رقمين عشريين يمثلان القوة من 2 .

أمثلة :

```
11.110E+32B
101E-17B
```

يُرجى الانتباه إلى الحرف E !

أمثلة على التصريحات عن المتحولات الثنائية بفاصلة متحركة :

```
DECLARE KBN BINARY FLOAT(45), RA_X6
FLOAT BIN(10);
```

KBN بدقة تتألف من 45 رقماً ثنائياً .

RA_X6 بدقة تتألف من 10 أرقام ثنائية .

الدقة القصوى هي من 53 رقماً ثنائياً . الدقة بالغلط هي من 21 رقماً ثنائياً .

نلاحظ هنا إنه في جميع التصريحات السابقة ، قد جرى إهمال الخاصية REAL ، والسبب هو في كون لغة PL/ 1 تأخذها بالغلط .

ب ـ المعطيات المركبة

وتتألف من قسم حقيقي وقسم وهمي ، هذا الأخير ، كتابة، يُسبق بالحرف I .

أمثلة :

72.192I	DEC FIXED	وهمي
12E+23I	DEC FLOAT	—
11.101BI	BIN FIXED	—
101E-17BI	BIN FLOAT	—

الخاصية COMPLEX (مختصر CPLX) يجب أن تكون إلزامية وموجودة في التصريح عن المتحولة المركبة ، وإلا ستؤخذ هذه المتحولة كحقيقية بالغلط .

مثلاً :

```
DECLARE VECT COMPLEX DECIMAL FIXED
(10,4);
```

المتحولة VECT ستتألف من قسم حقيقي (10, 4) DEC1è,ewd FIXED وقسم وهمي (10, 4) DEC FIXED . بإمكان تعليمة الطباعة أن تعطي النتيجة التالية :

```
VECT = 54.6530 + 643.66781;
```

من المفهوم ، إذا كان VECT قد جرى التصريح عنها وكأنها CPLX فمن الممكن أن نكتب :

```
VECT = SQRT (expression);
```

نفس الشيء في حالة طباعة تعبير بقيمة سلبية .

2 ـ الصيغ بالغلط

في حالة المتحولات من النوع الجبري، وإذا لم يتم تحديد أية خاصية ، سيختار المصرِّف :

للمعرِّفات التي تبدأ بالأحرف : I, J, K, L, M, N	REAL BINARY FIXED(15,0)
للمعرِّفات التي لا تبدأ بهذه الأحرف	REAL DECIMAL FLOAT(6)

وإذا حدَّدنا على الأقل خاصية واحدة ، فالأنواع الضمنية المعتمدة ستكون :

REAL, DECIMAL et FLOAT.

مثلاً التصريحات التالية :

```
DECLARE IA FIXED, ID DECIMAL, FB FLOAT;
```

تعطي الخاصيات التالية

```
IA DECIMAL FIXED(5,0)
ID DECIMAL FLOAT(6)
FB DECIMAL FLOAT(6)
```

قمنا بإيجاز الخاصيات التالية :

الطريقة :	COMPLEX	REAL	بإيجاز	..	CPLX
القاعدة :	DECIMAL	BINARY	بإيجاز	BIN	DEC
المعيار :	FLOAT	FIXED	بإيجاز	.	..
الدقة :	(عدد الأرقام [، مُعامل المعيار])				

ملاحظة عامة حول كتابة الثوابت الجبرية :

لا يُمكن أن نضع أي فراغ في داخل الثوابت الجبرية ، وبشكل خاص ، الكتابة 0.1156E 6 + ليست مسموحة . يجب إذاً كتابة هذه الثابتة على الشكل التالي : 0.1156E6 + أو 0.1156E + 06

1.6.2 ـ السلاسل (Chaines)

هناك صنفان من السلاسل : سلاسل السمات وسلاسل البتات . وفي الحالتين ، يعادل طول السلسلة 767 32 سمة أو بتة .

أ ـ سلاسل السمات

يتعلَّق ذلك بسلسلة من السمات المختلفة ، بين أبوستروف .

أمثلة :

'ABCD' (طول السلسلة هو 4)

'NUMERO 5643' . (الطول يعادل 11)

'*' (الطول 1)

يتمثّل الأبوستروف بسلسلة من 2 أبوستروف ('') .

```
'NOM = ''JEAN-PIERRE'''
```

تنشيء السلسلة :

```
NOM = 'JEAN-PIERRE'
```

من الممكن أن نضع مُعامِل تكرار أمام السلسلة :

```
(15)'*'
```

وهي تعادل :

```
'***************'
```

معامِل التكرار هو دائماً عبارة عن ثابتة صحيحة عشرية إيجابية بين هلالين .
نصرِّح عن المتحولة وكأنها من نوع سلسلة من السمات حسب الطريقة التالية : (باختصار CHAR) ؛

```
DECLARE TEXTE CHARACTER(20);
```

وهـذا يعني إن المتحولـة TEXTE هي عبارة عن سلسلة من السمـات بطول 20 سمة .

ب ـ سلاسل البتات

الفرق الوحيد الموجود بينها وبين سلاسل السمات هو في كونها متبوعة بالحرف B وبأنها لا تحتوي سوى على أرقام ثنائية 0 و1 .

مثلاً :

```
'1110010'B
'0'B
'1'B
```

السلسلة B '1' (5) هي معادلة للسلسلة 'B'11111 .

نُصرِّح عن سلسلة البتات حسب الطريقة التالية :

```
DECLARE IDENT BIT(50);
```

المتحولة IDENT هي عبارة عن سلسلة من البتات .

2 ـ الخاصية VARYING

إذا كنا نرغب في أن تكون متحولة من نوع سلسلة بطول مُتحوِّل فسنحدِّد لها الخاصية VARYING (باختصار VAR) .

```
DECLARE TEXT_A CHAR(20) VARYING;
```

المتحولة TEXT-A هي بطول متحول ، وبطول قصوي يعادل 20 سمة .
يمكن أن تكون سلسلة السمات بطول صفر .

1.63 ـ إعداد المعطيات

من الممكن إعداد متحولة في تصريح ، بواسطة الخاصية
نضع بين هلالين ، خلف الكلمة ـ مفتاح INITIAL ، ثابتة من نفس نوع المتحولة المطلوب إعدادها .

أمثلة :

```
DCL X DEC FIXED INITIAL(0);
```

المتحولة X ، عشرية بفاصلة ثابتة ودقة (5,0) بالغلط ، تصبح معادلة لـ 0 كقيمة أولية .

```
DCL VH DEC FIXED INIT(24);
```

المتحولة VH ستصبح معادلة لـ 24.

```
DCL IMC BIN FIXED INIT(10011B);
```

المتحولة IMC ستصبح 10011B

```
DCL TEXT CHARACTER(50) VARYING
INITIAL('*');
```

المتحولة TEXT ، سلسلة بطول قصوي يعادل 50 سمة ، وفي البداية تأخذ القيمة ' * ' والطول 1 .

1.64 ـ معطيات التحكّم بالبرنامج

يُصرَّح عن المعرِّف الذي يُمثل السمة على الشكل التالي :

```
DECLARE SUITE LABEL;
```

من الممكن أيضاً تجميع التصريحات :

```
DCL (A,B,C,D,E,F) LABEL;
```

ولكن في أغلب الأحيان يُصرَّح عن الأوسمة نصاً :

```
BOUCLE: X = X + 1;
```

المعرَّف BOUCLE ، المتبوع بالسمة نقطتين (:) وهو عبارة عن وسم .

كما في بعض المعطيات ، يُمكن للأوسمة أن تكون موجودة في تركيبات أو مُنظَّمة في جداول (أنظر لاحقاً) .

من الممكن إستعمالها في تعليمات نظيرية :

```
SUITE = BOUCLE;

GO TO SUITE;
```

التعليمة GO TO SUITE ، ستنفَّذ فعلياً مثل ; GO TO BOUCLE .

1.7 ـ تعليمة التخصيص

تُمثِّل تعليمة التخصيص ، أو التنظيم بواسطة السمة =. وتُنفذ من اليمين إلى اليسار . ويُمكن أن تكون بسيطة أو مُتعدِّدة .

أمثلة :

```
A = X;
```
(تخصيص بسيط)

هناك إنتقال لمضمون X ، الذي يبقى بدون تغيير ، إلى المتحولة A .

```
A, B, C = Y;
```
(تخصيص متعدِّد)

هناك إنتقال لمضمونْ Y إلى المتحولات A, B, C .

2 ـ تركيبة المعطيات

2.1 ـ الجداول

2.1.1 ـ التصريح عن الجداول

الجدول يمكن أن يكون بـ 32 بعداً ، والمتحولات المناسبة بداخله ستكون بـ 32 دليلاً كحد أقصى .

ويُصرَّح عن الجداول على الشكل التالي :

```
DECLARE TAB(20) DECIMAL FIXED(6);
DECLARE XT(30,30) FIXED;
```

المتحولة TAB تمثّل جدولاً حيث الدليل فيه يسير من 1 إلى 20 ، أي ببعد واحد ،

بينما XT هي عبارة عن جدول ببعدين . جميع عناصر TAB هي من نوع DEC FIXED (6, 0) ، بينما جميع العناصر في XT هي من نوع DEC FIXED (5, 0) .

حدود الأبعاد هي من 1 إلى 20 للجدول TAB ، ومن 1 إلى 30 لكل واحد من أبعاد XT .

من الممكن في لغة PL/ 1 تثبيت حدود الأبعاد في قيم مختلفة . هكذا ، فالتعليمة :

```
DECLARE TAB(20) ...;
```

من الممكن أيضاً الكتابة، وبشكل كامل .

```
DECLARE TAB(1:20) ...;
```

البعد الذي يتمتع بحدود أقل من 1 هو نموذجي . ومن الممكن تحديد أبعاد الجدول TAB .

```
DECLARE TAB(-5:14) ...;
```

في لغة PL / 1 ، تُرتب الجداول في الذاكرة حسب الدلائل المتصاعدة لجهة اليمين . هكذا ، فالجدول X (10, 10) يتم ترتيبه على الشكل التالي :

```
X(1,1)  X(1,2)  X(1,3)  X(1,4)  ...
                                X(1,10)
X(2,1)  X(2,2)  ...
```

2.1.2 ـ إعداد الجداول

بالإمكان إستعمال الخاصة INITIAL لإعداد وتهيئة الجداول . من الممكن إعداد الجدول كلياً أو جزئياً .

مثلاً :

```
DECLARE CS(100) DEC FIXED INITIAL
                            ((100)0);
```

في هذه الحالة يتم إعداد الجدول كلياً في صفر ، بواسطة المُعامِل (100)0 في INITIAL .

في الحالة التالية :

```
DECLARE HH(100) DEC FIXED INIT
              ((25)1, (25)0, (50)-1);
```

هنا يجري إعداد الجدول HH كلياً ، فالعناصر الـ 25 الأولى تأخذ القيمة 1 ، وال 25 التالية تأخذ القيمة 0 أما الباقي (50 عنصراً) فتأخذ القيمة 1 -

ومن الممكن إعداد الجدول جزئياً :

```
DECLARE LIIK(10) CHAR(5) INIT
                          ((2)(5)'  ');
```

سيتم إعداد أول عنصرين من هذا الجدول في حيِّزين من 5 فراغات. 'b' (5) تعادل 'bbbbb'.

2.1.3 ـ معالجة الجداول

هناك ثلاث طرق لمعالجة الجداول :

- بواسطة العناصر مع تحديد جميع الدلائل (index).
- بالأسطر أو الأعمدة.
- في المجموعة.

في لغة PL/1 ، يمكن إستعمال أي تعبير مُحوَّل إلى قيمة جبرية صحيحة كدليل.

مثلاً :

```
DECLARE (A(10,10), B(10,10),
              C(10,10)) ...;
```

- بواسطة العناصر

```
B(1,2) = A(1,1) + C(2,3)*3;
```

- بالأسطر أو بالأعمدة

نستعمل الدليل * ، الصالح لجميع عناصر السطر (أو بشكل عام للبعد بكامله).

```
B(*,1) = A(*,2) + C(*,3);
A(5,*) = 0;
C(*,10) = C(*,10)*2;
```

مثلاً ، التعليمة الثانية تعني إن جميع العناصر من البعد الثاني من A يجب أن تأخذ القيمة 0.

- في المجموعة

لا نحدِّد الدلائل ، مما يعني إن جميع العناصر مستعملة

```
A = C;
B = A + C;
A = 0;
A = B*C;
```

جميع العمليات تتم لعنصر بعد عنصر ، لكامل الجدول .الجداول يجب أن تكون بأبعاد بنفس الطريقة .

2.2 ـ التركيبات

2.2.1 ـ تعريف الرقم العشيري (التراتبي)

كما في لغة كوبول ، فتركية PL / 1 هي عشيرية بواسطة الأرقام .

أمثلة :

```
DECLARE 1 ALPHA,
          2 BRAVO,
            3 CHARLIE ...,
            3 DELTA ...,
            3 ECHO ...,
          2 FOX,
            3 GOLF ...,
            3 HOTEL ...;
```

وكما في لغة الكوبول أيضاً ، ليس من الضروري أن تكون الأرقام التراتبية متتابعة بشكل مباشر .

هذه المواضيع هي مستعملة غالباً وكما في كوبول ، لوصف التسجيلات . نجدها إذاً في تعليمات الإدخال ـ والإخراج .

2.2.2 ـ تخصيص التركيبات

أ ـ الحالة العادية (التركيبات المتشابهة)

لنفترض تركيبتين .

```
DECLARE 1 A,
              2 B,
                   3 C,
                   3 D,
              2 E,
                   3 F,
                   3 G;
DECLARE 1 H,
              2 I,
                   3 J,
                   3 K,
              2 L,
                   3 M,
                   3 N;
```

التعليمة A = H هي معادلة للتعليمات :

```
C = J;
D = K;
F = M;
G = N;
```

بنفس الطريقة ، التعليمة A = H** 2 / A; هي معادلة للتعليمات :

```
C = J**2/C;
D = K**2/D;
F = M**2/F;
G = N**2/G;
```

ب ـ الحالة BYNAME (في كوبول : CORRESPONDING)

من الممكن تحديد تعليمات تخصيص التركيبات للعناصر المنتهية بنفس الاسم . كل سلسلة من معرِّفات الأعداد بمستوى أعلى من 1 يجب أن تكون نفسها في كل حالة .

لنفترض تركيبتين :

```
DECLARE 1 A, 2 B,
               3 C,
               3 D,
             2 F LIKE B,
        1 X, 2 B,
               3 Z,
               3 D,
             2 F,
               3 C, 3 Z;
```

التعليمة :

```
A = X, BY NAME;
```

هي مُعادلة للتعليمتين :

```
A.B.D = X.B.D;
A.F.C = X.F.C;
```

من الجهة الأولى أو الثانية للاشارة = ، يجب أن تكون الأسماء المركّبة هي نفسها ما عدا المعرِّف بالمستوى 1 .

3 ـ التعابير (EXPRESSION)

هناك ثلاثة أصناف من التعابير :

ـ بسيطة

ـ جداول
ـ تركيبات
مثلاً :

```
DECLARE A(10,10) BIN FIXED(30),
        B(10,10) BIN FIXED(30),
        1 TAUX,
              2 PRIM DEC FIXED(6,2),
              2 SECD DEC FIXED(6,2),
        1 COUT LIKE TAUX,
        (C,D) BIN FIXED(15);
```

A وB هي عبـارة عن جداول ، TAUX وCOUT هي تـركيبات ، C وD هي متحولات بسيطة .

تعابير بسيطة :

```
C*D;
A(3,2) + B(4,8);
TAUX.PRIM - COUT.PRIM;
TAUX.SEC/4 + COUT.SEC*A(5,5);
```

تعابير جداول :

```
A + B;
A*C - D;
B/10B;
```

(نضرب جميع عناصر A بواسطة C ونطرح منها في كل مرّة القيمة C D)

(10B هي ثابتة ثنائية)

تعابير تركيبات

```
TAUX * COUT;
TAUX/2;
```

3.1 ـ المؤثرات (OPERATORS)

هناك عدة أصناف من المؤثرات في لغة PL/ 1 :

ـ مؤثرات جبرية .
ـ مؤثرات في سلاسل البتات .
ـ مؤثرات المقارنة .
ـ مؤثرات الإلتحام .

هذه المؤثرات تخضع لقواعد الأفضلية ، التي يُمكن إجتيازها بواسطة الأهلّـة . تُرتّـب المؤثرات حسب الأولوية المتناقصة في اللائحة التالية :

** (exponentiation) + - ¬ (لا)

* (ضرب / قسمة)

+ جمع

|| إلتحام

(مؤثرات للمقارنة) < ¬< <= = ¬= >= > ¬>

& (« و » قاصرة) («et» exclusif)

| (« أو » مشتملة «OU» inclusif) .

3.1.1 ـ مؤثرات جبرية

هناك فرق بين - و + prefixes ومؤثرات العلاقة الجبرية الجمع والطرح + - . المؤثرات prefixe تُنفَّذ دائماً بأفضلية .

من الممكن أن نكتب إذاً في لغة PL/ 1 :

A * - B الذي يعني A * (- B)

إضافة لذلك يُمكن أن يظهر عدة مؤثرات prefixe على التوالي :

A - تعني A بإشارة معكوسة .

A - - تعني تغيير إشارة A مرتين ، إذا A + .

A - - - تعني تغيير إشارة A ثلاث مرات إذا A - ، الخ .

التحويلات

التعابير المختلطة هي مسموحة في لغة PL/ 1 ، وبشكل عام ،تعمل PL/ 1 كثيراً لتنفيذها ، طالما تحتفظ بمعنى جبري . وإلا ، هناك تحريك للشرط CONVERSION مع صيغة نموذجية أو غير نموذجية ، حسب إدارة الانقطاع المُختار .

تُحدَّد الأنواع ، القاعدة ، الطريقة ، الدقة ، والمعيار بواسطة المتحولة « الهدف » ، إلى يسار الإشارة = .

في الحالة التي يوجد فيها عدم تجانس ، تتم التحويلات حسب القواعد المحدَّدة . وفي الحالة التي يكون فيها التعبير متجانساً ، يتم الحساب بشكل عام في الطريقة ، القاعدة والمعيار المعتمد .

تُوجز قواعد التحويل كما هو أعلاه :

ـ النوع TYPE : إذا كانت سلسلة السمات بمعنى جبري . كما في المثل '123.45' ، فالتحويل هو قائم ، هنا الى DECIMAL FIXED . وإذا لم يكن للسلسلة أي معنى جبري ، مثلاً 'AR56,6TH' ، فالشرط CONVERSION هو متحرِّك .

ـ قاعدة Base : إذا كان هناك فروق في القاعدة في تعبير مختلط ، فسيحوًل DECIMAL إلى BINARY .

ـ الطريقة MODE : إذا كان هناك إختلاف في الطريقة ، فسيحوًل REAL إلى COMPLEX ، مع قسم وهمي صفر .

ـ المعيار Echélle : إذا كان هناك فروقات في المعيار ، فسيحوًل FIXED إلى FLOAT .

3.1.2 ـ مؤثرات سلاسل البتات (المنطقية)

المؤثرات هي :

¬ & |

تتم العمليات بتة بعد بتة . وتنتج العملية سلسلة جديدة من البتات ، تتطابق مع السابقة .

لنفترض السلاسل :

```
A = '010111'B;
B = '111111'B;
C = '110'B;
```

المؤثر « عاكس » (لا) (¬) يعكس البتات : فالبتة '0' B تصبح '1' B والعكس .

```
¬ A → '101000'B
¬ B → '000000'B
```
الخ .

المؤثر « و » (&) يؤدي إلى جمع منطقي قاصر :

```
'1'B & '1'B → '1'B
'1'B & '0'B → '0'B
'0'B & '1'B → '0'B
'0'B & '0'B → '0'B
```

هكذا :

```
A & B → '010111'B
C & B → '110000'B
```

ملاحظة :

التسطير يتم من الشمال ونستكمل السلسلة الأقصر مع البتات '0' B لجهة اليمين .

المؤثر « أو » (|) يؤدي إلى إتحاد (جمع منطقي مشتمل)

```
'1'B | '1'B → '1'B
'1'B | '0'B → '1'B
'0'B | '1'B → '1'B
'0'B | '0'B → '0'B
```

هكذا :

```
A | B → '111111'B
C | B → '111111'B
```

من الممكن أيضاً تجميع المؤثرات :

```
A | ¬ C → '011111'B
```

ومن الممكن أيضاً وضعها بالتسلسل :

```
A & B | C → '110111'B
```

3.1.3 ـ مؤثرات المقارنة

هناك ثلاثة أنواع من المقارنات :

- الجبرية : إذا كانت المتأثرات تختلف بالخاصيات ، فستحوَّل حسب القواعد المذكـورة المناسبة للتحويلات .
- سلاسل السمات : المقارنة تتم من اليسار إلى اليمين ، لسمة بعد سمة .
- سلاسل البتات : المقارنة تتم من اليسار إلى اليمين وبتة بعد بتة . في الحالتين الأخيرتين ، ولسلاسل بطول مختلف ، نتوسع قليلاً إلى اليسار مع فراغات أو أصفار ثنائية .

المؤثرات هي :

```
<  =  >  <=  >=  ¬<  ¬>  ¬=
```

في جميع الحالات ، نتيجة المقارنـة هي سلسلة من البتات بـطول 1 . قيمة هـذه السلسلة من البتات :

'1'B ⟶ حقيقة (TRUE)

'0'B⟶ غلط (FALSE)

3.1.4 ـ الإلتحام (Concaténation)

مؤثر الإلتحام هو :

```
||
```

الإلتحام في تعبير رياضي يؤدي إلى تحويل في نوع سلسلة السمات (إذا كان هناك عدم تجانس) .

الإلتحام يستخدم لتوافق سلسلتين من السمات أو من البتات .

لنفترض :

```
A  =  '010111'B;     سلسلة من البتات
B  =  '101'B;         —
C  =  'XY,Z';        سلسلة من السمات
D  =  'AA/BB';        —
A||B → '010111101'B
A||A||B → '010111010111101'B
C||D → 'XY,ZAA/BB'
D||C → 'AA/BBXY,Z'
B||D → '101AA/BB'   ( تحويل )
```

الخاصية VARYING . تستخدم للدلالة على إن سلسلة البتات أو السمات هي بطول متحول .

من الممكن أيضاً تشكيل فقرات أو فصول بواسطة الإلتحام المتتالي لقطع من النصوص على متجولة مصرَّح عنها بأنها من نوع VARYING . طول المتحولة يتغير بعد كل عملية التحام ، ولكي لا يزيد عن طول معين قصوي موجود في التصريح (في المثل أعلاه : 3000) .

مثل عن برنامج كامل يؤلف نصاً :

```
EXEPL: PROCEDURE OPTIONS(MAIN);
       DECLARE TEXTE CHARACTER(3000)
                          VARYING
                          INITIAL(' ');
       DECLARE TO CHAR(78), NC DECIMAL
                                  (2);
       LECTURE: GET EDIT(NC,TC) (F(2),
                                A(78));
       IF NC=99 THEN GO TO FIN;
       TEXTE = TEXTE||TC; GO TO
                              LECTURE;
       FIN: PUT DATA(TEXTE);
END EXEPL;
```

4ـ التحكّم ، مراقبة البرنامج (PROGRAM CONTROL)

أوامر التحكم بالبرنامج هي التالية :

```
GO TO  GOTO
IF
DO
CALL
RETURN
END
STOP
EXIT
```

GO TO - I

أمر التفريع غير المشروط إلى وسم معين ، لنشر إلى أنه يجب أن يكون هناك فراغ واحد على الأقل بين معرِّف الوسم .

مثلاً :

```
GO TO INSTR_65;
```

IF - II

هذه التعليمة يُمكن أن تأخذ شكلين :

```
(1) IF expression-élémentaire THEN unité_1 ;
(2) IF expression-élémentaire THEN unité_1 ;
                              ELSE unité_2 ;
```

expression - elementaire - هو عبارة عن تعبير رياضي جبري أو منطقي أو متحولة . < unité - 1 > و < unité - 2 > هي أوامر بسيطة ، أو مجموعة DO ، أو قدرات BEGIN .

هناك سبعة أوامر بسيطة ممنوعة بعد THEN أو ELSE .

وهي (1) :

```
DO | END | PROCEDURE | BEGIN | DECLARE
                         | FORMAT | ENTRY
```

مثل 1 :

```
IF A+B=C-D THEN X=Y; ELSE X=Z;
```

إذا (A + B = C - D) هو حقيقة نقوم بتنفيذ «THEN» ، أي X = Y ، ونسير إلى التعليمة التالية < Instr.suiv. > بالقفز عن «ELSE» .

وإذا (A + B = C - D) هو غلط نقفز عن «THEN» ونقوم بتنفيذ «ELSE» أو Z = X ، ونسير إلى التعليمة التالية .

مثل 2 :

```
IF A+B>C+D THEN GO TO SUITE; < instr. suiv. >;
```

إذا كان (A + B > C + D) هو حقيقة ، يوجد تفريع إلى SUITE . وإلا ، تُهمل THEN ويجري العبور إلى التعليمة التالية .

---

(1) بالنسبة لـ DO ، يتعلّق ذلك بالأمر DO التكراري .

من الممكن تداخل الأوامر IF ، ولكن هذا لا ينصح به : هكذا : فالأوامر IF المتداخلة يمكن أن تشكل منبعاً للأخطاء المنطقية الخاصة والمهمة .

DO - III

هناك نوعان من هذا الأمر

- الأمر DO غير المتكرّر.
- الأمر DO التكراري .

DO - A غير المتكرر أو المجموع

```
DO; ...; ...; ...; ...; END;
```

المجموعة DO تسمح بمعالجة سلسلة من التعليمات كوحدة كاملة . وتستعمل بشكل أساسي خلف THEN أو ELSE في تعليمة IF .

DO - B التكرارية

1 - حلقة بسيطة

الشكل العام للحلقات هو التالي :

```
DO variable = expression1 TO expr2 BY expr3 ;
.
.
END;
```

التعابير Expression يجب أن تكون تعابير لا إتجاهية . الاختيار يتم في بداية الحلقة .

عندما يجري إلغاء BY ، يأخذ المصرّف بالغلط BY 1 .

عندما نستعمل تعابير في موقع A ، B أو C ، يتم حساب هذه التعابير قبل الدخول الى الحلقة . تُخزّن النتائج في متحولات EX3, EX2, EX1 ، ينشئها المُصرّف لهذه الغاية ، ولها نفس خاصيات التعابير المحسوبة .

إضافة لذلك ، يُمكن تبديل BY وTO بدون صعوبة . النتيجة النهائية هي نفسها لتنفيذ الحلقة .

إذا جرى إهمال TO ، فستدور الحلقة طويلاً طالما لا يؤدي إختبار البرنامج أو الانقطاع فيه إلى وقفها .

وفي النهاية ، إذا جرى إهمال TO وBY في نفس الوقت ، فلن تقوم الحلقة بالدوران سوى لمرة واحدة ، وذلك بالقيمة المُخصّصة لمتحولة الحلقة في بداية التعليمة DO .

2 ـ التكرار المتقطع

أمثلة :

```
DO BETA = 0 TO 50 BY 5, 100 TO 1000 BY
                                     10;
...
END;
```

ستدور الحلقة لقيمة BETA من 0, 5, 10...50 ، وبعد ذلك للقِيم 1000..., 120, 110, 120 .

3 ـ التكرار المشروط

من الممكن إنشاء حلقات تدور طالما إن التعبير الشرطي هو على القيمة « حقيقة » (أو 'B'1) .

نستعمل لذلك التعليمة WHILE .

الشكل العام هو :

```
DO   WHILE(expr. cond)
.
END;
```

تُحسب قيمة التعبير الشرطي البسيط «expr. cond» في بداية الحلقة وعند كل عبور لها . وطالما إن هذه القيمة هي 'B'1 ، فالحلقة سيجري تنفيذها . وعندما تصبح 'B'0 (غلط) ، يرجع التحكم إلى التعليمة التي تتبع END في نهاية الحلقة .

من الممكن خلط مختلف أشكال الحلقات كما نرغب .

أمثلة :

```
DO FRS = 85 TO 600, 0 TO 1000   WHILE
                                 (X <0);   (*)
...
END;
DO CAN=0 BY 10  WHILE(S*S >= EPS);
...
END;
```

في الحالة الأخيرة ، تأخذ المتحولة CAN القيم المتتالية 0 ، 10 ، 20 ، 30 ، الخ . بدون حدود مسبقة . تقف الحلقة عندما يصبح S * S أقل (<) من EPS .

(*) WHILE لا تحمل إلا على المجموعة الثانية ، من 0 إلى 1000

القواعد المتعلقة بانتقال التحكم للمجموعات DO .

من الممكن دائماً الخروج في أي نقطة من المجموعة DO ، التكرارية أولاً .

من الممكن الدخول في أي مكان في المجموعة DO غير التكرارية، بواسطة GOTO خارجة عن هذه المجموعة . ونظرياً ، هذا ممنوع لمجموعة DO تكرارية (حلقة DO) . لا يمكننا الدخول إلى الحلقة إلا في حدود الأمر DO نفسه .

CALL - IV

تستعمل هذه التعليمة لدعـوة إجراء (procedure) معـين ، أي لتحريكـه ونقل التحكم إلى نقطة الدخول إلى هذا الإجراء .

ملاحظة : لا يُنفَّـذ الأجراء أبداً على التوالي؛ يجب دعوته دائماً بواسطة CALL .

مثلاً :

```
CALL PROC_6;
```

RETURN - V

هذه التعليمة تنهي إجراء معين وتعيذ التحكم إلى النقطة التالية لـ CALL والمناسبة للأجراء . في غياب الأمر RETURN في أحد الإجراءات ، سيكون لـ END في نهاية الأجراء نفس الفعل .

في حالة الدوال ، بإمكان RETURN أن تعيد قيمة معينة . وتستعمل عادة بالشكل التالي :

```
RETURN (expression);
```

هذا الإستعمال سيظهر لنا خلال دراسة الدوال .

END - VI

ينهي دائماً الفدرات أو المجموعات . END يُمكن أن تكون متبوعة بواسطة وسم المجموعة أو الفدرة (BEGIN أو PROCEDURE) . عدة أوامر END متتالية يُمكن أن تتطابق في واحد . عندما يبلغ التحكُّم النقطة END المناسبة لفدرة معينة BEGIN أو PROCEDURE ، تصبح هذه الفدرة غير فعالة . والتحكم يتتابع على التوالي (حالة BEGIN) أو يعود خلف CALL المناسبة (حالة الأجراء) .

EXIT و STOP — VII

STOP ينهي البرنامج كاملاً . والتحكم ينتقل إلى النظام . أما EXIT فينهي عملاً معيناً أو جميع الأعمال الموجودة ، في محيط أو نظام مُتعدِّد البرمجة ـ وفي برنامج عادي ، يقوم

STOP وEXIT بنفس الفعل .

وإذا لم يكن STOP موجوداً في البرنامج ، فمهمة الإنهاء الفعلي للبرنامج تقع على عاتق END الموجودة في نهاية الإجراء المركزي ((OPTIONS (MAIN)) .

ملاحظة مهمة جداً

يجب تفادي إنهاء ابرنامج بواسطة STOP أو END . هذه الإنهاءات تُدعى « غير عادية » وتُعيد إلى النظام كوداً ـ شرطياً لنهاية غير عادية للبرنامج . أو إذا كان العمل كاملاً يحتوي على عدة مراحل ، فالمرحلة الداخلية يُمكن أن نقفز عنها ، والنظام سيفحص الأكواد الشرطية قبل كل مرحلة .

النهاية الوحيدة الطبيعية للبرنامج تتم بواسطة END ، آخر تعليمة من البرنامج . هذه «END» تحتوي على إسم الأجراء المركزي . مثلاً ؛ END PA .

لا نستعمل STOP إلا عندما نرغب بإثارة إنهاء غير عادي للبرنامج .

مثلاً :

```
ON CONVERSION STOP;
```

5 ـ إدخال ـ إخراج

5.1 ـ رسائل على قنصلة التحكُّم

(1) DISPLAY (chaîne de caractères) ;

(2) DISPLAY (chaîne) REPLY (variable) ;

في الحالة الأولى ، تُطبع سلسلة السمات بشكل بسيط وواضح على قنصلة التحكُّم .

في الحالة الثانية ، يقف البرنامج في إنتظار الجواب .

5.2 ـ إدخال ـ إخراج مع تحويل ، بسيط

الكلمات المفاتيح الأساسية هي :

للادخال عند القراءة GET

للإخراج عند الكتابة PUT

عمليات الإدخال / الاخراج البسيطة (STREAM) هي ثلاثة أنواع :

ـ موجهة بواسطة المعطيات DATA

ـ موجهة بواسطة لائحة LIST

- مع تحكّم بالتنقيح EDIT

أ - عمليات الإدخال / الإخراج المُوجهة بواسطة المعطيات

GET DATA (liste de variables) ;
PUT DATA (liste de variables) ;

Liste de variable - لائحة بالمتحولات

عند الإدخال : ٣تحضر المعطيات بالشكل العام :

Identificator - 1 = valeurs, identificateur - 2 = valeurs, ...,

معرّف - 2 = قيم معرّف - 1 = قيم

عند الإخراج : تُحضر التعليمة على الشكل التالي :

حيث لا يُمكن أن تكون D, C, B, A سوى متحولات بسيطة ، أو جداول ، أو تركيبات، أو عناصر مؤشرة من جداول، أو عناصر مُميَّزة بتركيبات، بداخل نفس اللائحة ، من البديهي إستعمال مختلف أنواع هذه المتحولات بالتزامن .

والنتيجة ستحضر على الشكل التالي :

Identificateur = valeur ident - 2 = valeur ...

التعليمة ;PUT DATA بدون لائحة بالمتحولات تؤدي إلى طباعة جميع المتحولات (اللا اتجاهيــة ، الجـداول ، والتركيبات) ، المصرَّح عنها ، والمعروفة في لحظة تنفيذ التعليمة . الشكل العام هو نفسه كما في الحالة السابقة : معرّف = قيمة

(identificateur = valeur)

ب - الإدخال / الإخراج الموجهة بواسطة لائحة

هذا النوع من الإرسال هو شبيه بالنوع السابق ، مع الفرق التالي : إسم المعرّفات هو غير محدَّد أبداً ، لا في معطيات الإدخال ، ولا على مطبوعة الإخراج . اللائحة يجب أن تكون دائماً موجودة في تعليمة الإدخال :

GET LIST (liste de données) ;

اللائحة يجب أن تكون دائماً موجودة في تعليمة الإخراج .

PUT LIST (liste de données) ;

تحتوي اللائحة ، عند الإدخال ، على متحولات لا إتجاهية (Scalaire) ، أو جداول أو تركيبات ، أو عناصر مؤشرة من جداول ، أو في النهاية عناصر من تركيبات . عند الإخراج ، فهي أكثر عمومية ، وتحتوي إضافة إلى المتحولات ، على تعابير مختلفة (إذاً

وبشكل خاص على ثوابت) . عند الإدخال : تُحضَّر المعطيات على شكل لائحة من الثوابت ، مفصولة بواسطة فواصل أو بواسطة فراغات . يُمكن للائحة أن تنتهي بواسطة نقطة فاصلة ، بشرط أن تكون مفصولة عن آخر ثابتة بواسطة فراغ واحد على الأقل .

عند الإخراج : المعطيات تكون على شكل لائحة من الثوابت ، مفصولة عن بعضها بواسطة فراغات . لا يوجد في هذه الحالة إشارة إلى إسم المتحولة .

```
PUT LIST(A, B, C, D);
```

النتيجة :

```
.005 6.400 .200 -36.500
```

من المكن وضع تعابير رياضية في اللائحة عند الاخراج .

```
PUT LIST(A, A*B/C, (X+Y)**2);
PUT LIST((REC(K) DO K=1 TO 50));

                        PUT LIST( )
(REC(K) DO K=1 TO 50)

PUT LIST('RACINE DE', X, 'EGALE',
                     SQRT(X));
```

ج - إدخال / إخراج مع تحكم بالنشر

هذا النوع من الإدخال / الإخراج هو معادل للادخال / الإخراج في لغة فورتران . يوجد إستعمال لنسق مشابه ..

تُحضَّر التعليمة في الشكل العام :

اللائحة هي شبيهة باللائحة المستعملة في الحالة «LIST» . من الممكن إذاً استعمال متحولات فيها ، ثوابت وتعابير رياضية .

النسق هو عبارة عن لائحة من المواصفات ، مفصولة بواسطة فواصل ، كما في لغة فورتران (الانتباه إلى الفرق البسيط بينهما) .

- المواصفات في DECIMAL FIXED

F (*w*)	W هو عدد السمات في حيز الإصدار
F (*w*, *d*)	W هو العدد الكامل للسمات ، d هو عدد السمات بعد الفاصلة العشرية .
F (*w*, *d*, *p*)	p هو مُعامِل المعيار الذي يضرب فعلياً قيمة المعطيات الخارجية بواسطة 10^P

ـ المواصفات في DECIMAL FLOAT

E (w, d, p)   W هي العدد الكامل للسمات

d عدد السمات بعد الفاصلة العشرية .

p مُعامِل المعيار (إختياري)

ـ مواصفات السلاسل

B(w)   w هو طول سلسلة البتات

A(w)   w هو طول سلسلة السمات

في هاتين الحالتين ، w هو إختياري .

ـ مواصفات الفسحات

X(w)   نقفز عن w من السمات

هذه المواصفة لا يُمكن أن تستعمل كمواصفة أخيرة للنسق ، وسيتم إهمالها .

ـ مواصفات من نوع PICTURE

تُمثّل image ' P سلسلة من أكواد السمات PICTURE .

ملاحظة : في جميع الحالات ، d, w وp يُمكن أن تكون عبارة عن تعابير رياضية (ثوابت ، متحولات أو تعابير) .

مُعامِل التكرار

مُعامل التكرار هو بشكل عام عبارة عن ثابتة صحيحة إيجابية ، مفصولة عن مواصفة النسق بواسطة فراغ واحد على الأقل .

5A (25) تعادل A(25) ، A(25) ، A (25) ، A (25) ، A(25) .

من الممكن إستعمال أي تعبير كمعامل تكرار ، ولكن يجب أن يكون بداخل هلالين :

(X+Y/XK) A(25) الخ . .

د ـ التصفيح (ترتيب في صفحات)

عند الإخراج ، بتصرفنا مواصفات إضافية :

للتركيز على صفحة جديدة PAGE

SKIP (expression)

تحوّل expression إلى قيمة صحيحة w :

w > 0 : نقفز عن w سطر (أي ننشيء 1 - w سطراً أبيضاً)

w = 1 : SKIP = SKIP (1) : نرجع إلى السطر ،

w <= 0 : يوجد طباعة مكثفة

LINE (expr) (يُحوِّل التعبير expr إلى عدد صحيح w. ويوجد تركيز على السطر رقم w من الصفحة . إذا كان قد تم كتابة w سطر على الأقل ، أو إذا كان w أعلى من عدد الأسطر المتوقع لكل صفحة (بشكل عام 60) ، فسيؤدي ذلك إلى تحريك للشرط ENDPAGE (من هنا بشكل عام قفز عن صفحة) .

COLUMN (expr) تركيز على العامود رقم w . وإذا كانت قد تمت كتابة w سمة، فسنمرُّ إلى السطر التالي . يستعمل فقط في لائحة بنسق .

5.3 ـ المواصفات PICTURE

المواصفة PICTURE (أو صورة) تقوم على سلسلة من أكواد ـ السمات (أو السمات ـ صورة) التي تدل على عمليات التنقيح المطلوب إجراؤها على سلسلة من السمات .

هناك نوعان من المواصفات ـ صورة ، والتي تستعمل لوصف :

ـ سلاسل مختلفة من السمات .

ـ سلاسل من السمات رقمية .

تُوضع أكواد السمات التي تؤلف مواصفة ـ صورة بداخل أبوستروف وتستعمل إما مع الخاصية PICTURE في التصريح ، وإما في نسق P في تعليمة إدخال ـ إخراج موجهة نحو التنقيح (EDIT) .

أمثلة

```
DECLARE TAUX PICTURE'99V.999';
GET EDIT(ALPHA) (P'XXXXX');
```

أ ـ وصف سلاسل السمات المختلفة

هناك ثلاثة أكواد ـ سمات ممكنة لهذه السلاسل العادية :

X الموقع المُضاف ويحتوي على أي سمة معروفة من المكنة .

A الموقع المعتمد لا يحتوي إلا على سمة أبجدية أو فراغ .

9 الموقع المعتمد لا يحتوي إلا على سمة عشرية أو فراغ ، لا يوجد سوى سمتي إدخال .

على الأقل سمة X أو Y ستظهر في سلسلة أكواد ـ السمات في المواصفة ـ صورة .

ب ـ وصف السلاسل الرقمية

هذه السلاسل هي تلك التي تمتاز بمعنى رقمي .

أمثلة :

```
'8765.89'
'1234'
'+675'
```

الخ

ولكن

```
'45A567'
'45.78.56'
```

هي مختلفة وبدون أي معنى رقمي .

لا يمكن أبداً للمواصفات PICTURE الرقمية أن تحتوي على أكواد ـ سمات X أو A .

يُستعمل كود ـ السمة 9 ، ولكن معناه هو مختلف : أرقام عشرية فقط بدون فراغات .

وبينما يتم التركيز في السلاسل العادية لجهة اليسار ، يتم التركيز والتسطير في السلاسل الرقمية ، على الموقع المُخصَّص للفاصلة العشرية . يُحدَّد هذا الموقع بواسطة الحرف U .

مثلاً :

```
PICTURE'999V99'
```

يوجد عدد من أكواد ـ سمات لهذا النوع من السلاسل .

1 ـ أداة مواصفة الأرقام والفاصلة العشرية

9 الموقع المعتمد يحتوي على رقم (وليس على فراغ) .

V الموقع المعتمد هو مُخصَّص للنقطة العشرية (تسطير) ولكن لا يوجد أي إدخال لـ « . » في هذا الموقع .

2 ـ إلغاء الأصفار

Z تستبدل جميع الأصفار التي هي بدون دلالة بواسطة فراغات . لا يُمكن أن تظهر في نفس الوقت مع « * » .

* تستبدل جميع الأصفار التي هي بدون دلالة بواسطة نجوم * . لا يُمكن أن تظهر في نفس الوقت مع Z .

Y تستبدل جميع الأصفار بدون إستثناء بواسطة فراغات .

3 ـ سمات الإدخال (Insertion caracters)

{ , . / } تؤدي شرطياً إلى إدخال السمة المشار إليها ' . أو / إلى الموقع المُحدَّد .

B تؤدي بدون شرط إلى إدخال فراغ .

4 ـ أكواد ـ سمات تنقيح الإشارة

S تؤدي إلى طباعة إشارة (+ أو −) في الموقع المشار إليه بواسطة موقع السمة S .

+ تؤدي إلى طباعة السمة + إذا كان العدد إيجابياً . وفي الحالة المعاكسة ، سيتم إخراج فراغ في موقع هذه السمة .

− تؤدي إلى طباعة إشارة − للعدد السلبي . وإذا كان العدد إيجابياً ، سيتم إخراج فراغ في موقع هذه الاشارة .

تُوضع هذه الأكواد ـ السمـات في رأس السلسلة PICTURE فقط ، وفي موقـع الاشارة العادي .

5.4 ـ الإدخال ـ الإخراج العام

1 - 4 - 5 - فتح واغلاق السجلات .

الفتح :

OPEN FILE (nom-de-fichier) liste) ;

الاغلاق :

CLOSE FILE (nom-de-fichier) ; . هو إسم السجل

اللائحة (Liste) ، فقط في حالة الفتح ، هي عبارة عن لائحة بالصيغ ، مُركبة من عدد من الخاصيات الموصوفة أعلاه ، إضافةٍ إلى خاصيات خاصـة ، لا يمكن أن تكون موجودة إلا في هذه التعليمة .

5.4.2 ـ خاصيات وصف السجلات (Attributs de description des fichiers)

أ ـ الخاصيات المتناوبة

1 ـ STREAM وRECORD

لوصف نوع الإرسال

STREAM يُعالج السجل كدفق متواصل من المعطيات وبشكل سمات . هذه هي الحالة الكلاسيكية للبطاقات أو الطابعات .

RECORD يُعالج السجل كسلسلة من التسجيلات المنطقية ، تحتوي كل تسجيلة على واحد أو عدة معطيات في شكل داخلي .

2 ـ INPUT , OUTPUT UPDATE

تحدِّد إتجاه إرسال المعطيات .

INPUT إدخال . حركة السجل نحو البرنامج ، أو قراءة السجل .

OUTPUT إخراج . برنامج نحو السجل ، أو كتابة .

UPDATE إستيفاء يومي . قراءة وكتابة .

3 ـ SEQUENTIAL وDIRECT

تحدِّد طريقة البلوغ ومجموعة المعطيات

SEQUENTIAL هناك بلوغ للتسجيلات المتتالية ، وذلك بالرجوع إليها بواسطة المواقع المناسبة لها .

المثل النوعي هو السجل على شريط مغناطيسي . لا يمكننا بلوغ التسجيلة بالرتبة N إلا بعد بسط جميع التسجيلات السابقة .

DIRECT هناك بلوغ مباشر إلى التسجيلة بواسطة مفتاح .
هذه الخاصية تؤدي إلى الخاصية KEYED . إضافة لذلك يجب أن يكون الحجم (Volume) ببلوغ مباشر .

4 ـ BUFFERED وUNBUFFERED

يوجد أو لا يوجد إنتقال للمعطيات بواسطة حيز معطيات وسيطي (حيِّز داريء أو BUFFER) .

أ ـ خاصيات الإضافة

1 ـ PRINT

للسجلات STREAM وOUTPUT فقط .

نصرِّح عن السجلات PRINT إذا كنا نرغب بطباعتها مباشرةٍ أو بعد ذلك . السمة البدائية (سمة القفز) هي محفوظة . نتفادى إذاً الأخطاء الجارية في القفز عن الصفحات المتعددة بواسطة كتابة «1» في العامود 1 .

2 ـ BACKWARD

للسجلات RECORD SEQUENTIAL INPUT فقط .
تُحدِّد البلوغ إلى شريط مغناطيسي بواسطة النهاية (بلوغ عكسي) .

3 ـ KEYED

للسجلات DIRECT أو RECORD SEQUENTIAL
تُستعمل في الحالة التي نُضيف فيها مفتاح إلى كل تسجيلة .

4 ـ EXCLUSIVE

تستعمل EXCLUSIVE للسجلات RECORD DIRECT UPDATE KEYED فقط .

وتدل على إن التسجيلات يُمكن أن تكون محمية خلال تنفيذ عمل معين ، لتفادي كون عمل متزامن آخر لا يرغب بالبلوغ في نفس الوقت (برمجة متعددة) .

5 ـ ENVIRONMENT

هذه الخاصية تُحدِّد المُميِّزات الخارجية لمجموعة المعطيات . وكما هي ، توضع بعيد عن لغة PL / 1 . لائحة الصِيغ هي مختلفة من نظام إلى آخر .

5.4.3 ـ إرسال بدون تحويل

القراءة

READ FILE (nom-de-fichier) INTO (variable);

للسجلات SEQUENTIAL ، INPUT أو UPDATE .

يجب أن تكون المتحولة بمستوى 1 ، بدون دليل .
تؤدي كل تعليمة إلى قراءة تسجيلة في المتحولة. الأوامر READ المتتالية تقوم ببسط متتالي للتسجيلات في هذه المتحولة .

READ FILE (nom-de-fichier) IGNORE (expression);

للسجلات المتتالية SEQUENTIAL ، INPUT أو UPDATE
يُحسب التعبير (expression) ويُحوَّل إلى عدد صحيح n . وإذا كانت n>0 نقفز عن n تسجيلة .

READ FILE (nom-de-fichier) SET (pointeur);

للسجلات SEQUENTIAL ، INPUT أو BUFFERED, UPDATE .
من الممكن بهذه الطريقة بلوغ التسجيلات مباشرة في حيِّز الإدخال («Buffer») . التعليمة تُركِّز المؤشر على هذا الحيِّز . يُمكن للمؤشر أن يُعتمد لواحدة أو عدة متحولات مؤشرة .

READ FILE (nom-de-fichier) INTO (variable) KEY (expression);

للسجلات DIRECT ، INPUT ، أو UPDATE
يُحسب التعبير (.exp) و، إذا كان ضرورياً ، يُحوَّل إلى سلسلة من السمات .
نستعمل هذه السلسلة بعد ذلك لإيجاد التسجيلة التي لها نفس هذا المفتاح مباشرة .

الكتابة

WRITE FILE (nom-de-fichier) FROM (variable);

للسجلات SEQUENTIAL ، OUTPUT
المتحولة يجب أن تكون بمستوى 1 بدون دليل .
تؤدي كل تعليمة إلى كتابة تسجيلة على السجل الفيزيائي . الأوامر WRITE المتتالية تكتب التسجيلات الواحدة بعد الأخرى .

WRITE FILE (nom-de-fichier) FROM (variable)KEYFROM (expression);

للسجلات SEQUENTIAL أو OUTPUT, DIRECT
يجب أن يكون الحجم (الأسطوانة أو الشريط) ببلوغ إنتقائي . يُحوَّل التعبير محتملاً ، إلى سلسلة من السمات .

هذه السلسلة من السمات هي مفتاح التسجيلة . وحسب الحالات (التي تتعلّق بالنظام المستعمل) ، هذا المفتاح هو مُسجَّل أو غير مسجل . يُمكن للتسجيلة أن تكتب في أي موقع من السجل ، مُحدَّد بواسطة المفتاح ، وبشكل مباشر . للسجلات بتنظيم دليلي (indexe de file) ، يجب أن تتم الكتابة على التوالي ، وحسب الترتيب الأبجدي للمفاتيح المُسجَّلة . وفي الحالة التي تكون فيها السجلات بتنظيم مباشر ، يحتوي المفتاح على قسم يُحدِّد رقم المكان (الذي يناسب العنوان) وعلى قسم هو عبارة عن المفتاح المُسجَّل -RE GIONAL (3)) و(REGIONAL (2) ؛ لا يُوجد مفاتيح مُسجَّلة في REGIONAL . (1)

REWRITE FILE (nom-de-fichier) FROM (variable) KEY (expression) ;

للسجلات UPDATE, DIRECT أو SEQUENTIAL

التعبير (expr) في KEY يتبع نفس القواعد المعتمدة في الحالة السابقة . وفي الحالة التي تكون فيها السجلات SEQUENTIAL ، يجب أن تكون التسجيلة مقروءة قبل أن تكون مكتوبة من جديد . إضافة لذلك ، يجب أن يكون الحجم ببلوغ إنتقائي . يجب أن تسبق القراءة مباشرة عملية إعادة الكتابة . في حالة السجلات ببلوغ مباشر DIRECT ، هذه القاعدة ليست إلزامية .

LOCATE variable FILE (nom-de-fichier) SET (pointeur) ;

للسجلات SEQUENTIAL ، OUTPUT ، BUFFERED

الصيغة SET (pointeur) هي دائماً إختيارية . يجب أن تكون المتحولة مُؤشرة . وإذا أهملت SET ، سيأخذ المُصرِّف بالغلط الصيغة SET (P) حيث P ستكون المؤشر المعتمد للمتحولة .

الأمر LOCATE يُعرِّف المتحولة على داريء (Buffer) الإخراج للسجل موضع السؤال . يجب أن تُملأ المتحولة بعد ذلك . هذا هو الأمر التالي LOCATE ، أو إحتمالاً التعليمة CLOSE FILE (nom-de-fichier) هي التي تؤدي إلى كتابة التسجيلة المُحدَّدة على السِّجل .

LOCATE variable FILE (nom-de-fichier) KEYFROM (expression)
SET (pointeur) ;

للسجلات BUFFERED, OUTPUT, SEQUENTIAL

الصيغة KEYFROM تتبع القواعد المُشار إليها أعلاه للتعليمة ... WRITE FILE KEYFROM...; . الصيغة SET ، إختيارية ، تتبع القواعد المشار إليها للتعليمة

السابقة . هذه التعليمة LOCATE ... KEYFROM لا تستعمل إلا للسجلات بتنظيم دليلي أو مباشر مُعالج على التوالي .

```
DELETE FILE (nom-de-fichier) KEY (expression) ;
```

UPDATE, DIRECT للسجلات

هذه التعليمة لا تمحو التسجيلات ، ولكنها تضع علامة خاصة على هذه التسجيلات . هذه العلامة هي عبارة عن ثابتة '11111111111'B في أول بايتة من التسجيلة . التعبير في الصيغة KEY يتبع القواعد المشار إليها في الأعلى . التسجيلات المحوةلا تعاد قراءتها في قراءة متتالية ما عدا للتسجيلات بتنظيم REGIONAL (1) .

6 ـ إدارة الانقطاعات

عندما تحدث بعض الحوادث الطارئة عند تنفيذ برنامج مكتوب بلغة PL/ 1 (أو غيرها) ، مثل OVERFLOW ، القسمة على صفر ، قراءة بطاقة ـ نظام كمعطيات ، الخ . . . عند ذلك يحدث إنقطاع . وبشكل عام ينتج عن الانقطاع طباعة لتشخيص إنتهاء العمل الجاري .

هذا هو صحيح لأغلب اللغات المتطورة ، حيث لا يستطيع المبرمج عمل أي شيء ضد هذه الانقطاعات ، سوى إعادة قراءة برنامجه وتعديله لإيجاد متى وأين حدثت هذه الحادثة الطارئة أو العطل في البرنامج .

وفي لغة PL/ 1 ، من الممكن إستبدال فعل النظام بأفعال أخرى ، محدَّدة في البرنامج . وبكلمة أخرى ، من الممكن التحكُّم ، بتنفيذ وإدارة الانقطاع .

6.1 ـ النسق العام

```
ON nom-condition unité-on ;
```

أمثلة :

```
ON ENDFILE(FDICT) GO TO SUITE_1;
ON ENDPAGE BEGIN; ... ; END;
```

Unité-on

تُحدِّد الفعل المطلوب تنفيذه في حالة حدوث الانقطاع من النوع المشار إليه في nom-condition (إسم الشرط) . مثلاً ، في حالة الانقطاع الناتجة عن نهاية السجل (ENDFILE) ، يجب الذهاب إلى التعليمات SUITE-1 . مثلاً آخر ، في حالة الانقطاع الناتج عن محاولة كتابة بعيداً عن آخر سطر من الصفحة ، سيجب تنفيذ الفدرة BEGIN .

هذه «Unité-on» هي :

- إما تعليمة بسيطة، وحيدة ، بدون وسم .
- إما فدرة BEGIN بدون وسم .

وبشكل خاص فالتعليمات المركبة (ON وIF) والمجموعات DO هي ممنوعة .

Unité-on particulieres

SYSTEM تُحدِّد العودة إلى الشروط النموذجية للنظام .

مثلاً :

```
ON ENDPAGE SYSTEM;
```

هذه التعليمة تحدِّد إنه « من الآن فصاعداً » ، وللانقطاعات «UDPAGE» معنا الصيغة SYSTEM (أي في هذه الحالة القفز عن صفحة) .

بدون عمليات (instruction nulle) ؛

6.2 ـ لائحة بالشروط

عادة نُرتِّب شروط ـ الانقطاع في ثلاثة طبقات حسب إستعمالها .

- لتحريك واضح بواسطة أمر .
- دائماً مُحرَّك ، وسكون ممكن .
- دائماً مُحرَّك ، من غير الممكن أن يكون ساكناً .

أ ـ شروط للتحريك بشكل واضح

هي التالية :

```
SIZE SUBSCRIPTRANGE STRINGRANGE CHECK
```

SIZE

تؤدي إلى إنقطاع إذا كان هناك قطع (بتر) للمعلومات لجهة اليسار عند الارسال .

SUBSCRIPTRANGE SUBRG

تؤدي إلى إنقطاع إذا كان هناك طلب لعنصر من جدول من خارج حدود أبعاد هذا الجدول .

STRINGRANGE

تراقب حسن إستعمال الدالة SUBSTR .

CHECK (liste)

اللائحة (Liste) هي لائحة بالأوسمة / أو بالمتحولات . وفي كل مرَّة سنمرّ فيها على

هذه الأوسمة ، أو عندما نَعدِّل مضمون هذه المتحولات ، سيجري طباعة هذه الأوسمة (معرِّف فقط) أو هذه المتحولات (بالشكل الذي ستعطيه فيه التعليمة PUTDATA) . سنحصل إذاً على أثر مسار دوران البرنامج .

أ ـ الشروط الفعالة دائماً ، والتي من الممكن إلغاء فعاليتها

هي شروط الحساب . وهي تصبح غير فعالة بواسطة أمر prefixe ، وهي التالية :

```
CONVERSION       إختصار :  CONV
OVERFLOW                   OFL
FIXEDOVERFLOW              FOFL
UNDERFLOW                  UFL
ZERODIVIDE                 ZDIV
```

نستعملها عادة في التعليمات ON ، كما في بداية هذا الفصل . وهي تصبح غير فعالة بواسطة أمر prefixe يحرِّك كإجراء ، فدرة BEGIN ، أو تعليمة جبرية وحيـدة . يتألف الأمر (prefixe) من NO ، متبوعة مباشرة باسم الشرط .

مثلاً :

```
NOCONVERSION, NOZDIV.
```

ج ـ الشروط الفعالة دائماً ، والتي من غير الممكن إلغاء فعاليتهـا . وهي الشروطالتالية

AREA	مراقبة الذاكرة
CONDITION	شرط إصطناعي
ENDFILE	نهاية سجل متتالي
ENDPAGE	نهاية الصفحة
ERROR	غلط خطير
FINISH	إنهاء البرنامج
KEY	خطأ في المفتاح (سجلات)
NAME	خطأ في الاسم (سجلات)
RECORD	إدخال / إخراج على السجلات RECORD
TRANSMIT	إرسال
UNDEFINEDFILE (UNDF)	سجل غير مُحدَّد

هناك طريقان لإنهاء إدارة الانقطاع . كـل وحدة «Unité-on» تتصرَّف بشكل مستقل . يتعلَّـق ذلك إما بفدرة BEGIN ، المحدَّدة بوضوح ، وإما بتعليمة بسيطة . في الحالة الأخيرة ، كل شيء يسير كما لو كانت التعليمة مسبوقة بالمواصفة .

```
. G:PROCEDURE;
  END G;
```

الإنهاء الطبيعي يتم بواسطة END نهاية فدرة BEGIN أو بواسطة END وهمية لنهاية الإجراء PROCEDURE G .

الإنهاء غير الطبيعي هو إنهاء بـانتقال التحكُّم إلى خـارج الفدرة «unité-on» ، بواسطة تعليمة GOTO .

بعض الانقطاعات هي خطيرة . إنهاؤها الطبيعي يؤدي دائماً إلى إنقطاع ERROR ، نفسه ، إذا انتهى طبيعياً ، يؤدي إلى إنقطاع FINISH .

مثلاً :

```
ON ERROR PUT DATA;
```
إنهاء طبيعي

```
'ON ERROR GO TO ETIEXTR;
```
إنهاء غير طبيعي

في حالة الخطأ الخطير في الإجراء الذي يحتوي على هذه التعليمة ، سيكـون معنا تشخيص ، متبوع بالانقطاع ERROR الذي سيؤدي إلى إنتقـال التحكُّـم الى التعليمة ETIEXTR .

7 ـ الإجراءات SUBROUTINE وFUNCTION

من الممكن إنشـاء إجـراءات من نـوع دالـة (FUNCTION) أو منهـاج ثـانـوي SUBROUTINE في لغة PL/ 1 . هذه الاجراءات هي مُتخصِّصة ، وفي هـذا المعنى هناك عبور للمعاملات عند طلبها .

إضافة لذلك ، الطلبات هي نوعين :

- ضمنية ، للاجراءات من نوع دالة (function)
- واضحة ، للاجراءات من نوع منهاج ثانوي (SUBROUTINE)

7.1 ـ الاجراءات SUBROUTINE

الإجراء SUBROUTINE هو إجراء يُحرِّك ويصبح فعالاً بشكل واضح بواسطة أمر من نوع CALL . الشكل العام لهذا الأخير هو :

CALL nom-entrée [(argument 1, argument 2, ...)] ;

nom-entrée هو إسم الأجراء SUBROUTINE .

عندما ينتهي الإجراء SUBROUTINE ، يعود التحكُّم إلى التعليمـة التي تتبع

مباشرة الأمر CALL .

هناك طريقتان أساسيتان لإنهاء طبيعي للأجراء ـ SUBROUTINE :

ـ التحكّم ، في الاجراء SUBROUTINE المطلوب ، يصل إلى التعليمة END في نهاية الأجراء .

ـ التحكّم ، في هذا الأجراء نفسه ، يبلغ تعليمة RETURN في موقع مُعين من الأجراء . هذه التعليمة تُستعمل في هذه الحالة فقط ، بدون لائحة .

7.2 ـ الأجراء FUNCTION

من جهة التحكّم ، تسير الأشياء بنفس الطريقة كما في الحالة السابقة SUBROUTINE .

الدعوة (طلب الدالة) هي ضمنية ، كما بالنسبة لدالة PL / 1 عادية :

مثلاً :

```
Z = FACT(N);
```

إعادة القيمة المحسوبة بواسطة الدالة يتم بواسطة أمر RETURN ، مع لائحة تحتوي على تعبير بسيط يعطي هذه القيمة . التعبير يمكن أن يكون ثابتة ، متحولة ، أو تعبير مُعقد كما نرغب :

<u>RETURN</u> (expression-scalaire) ;

ملاحظة : وكما يوجد دوال في لغة PL/ 1 ، تعيد الجداول (مثل SIN ، SQRT ، INDEX ، الخ وأغلب دوال الحساب) ، فالقيمة المعادة بواسطة إجراء ـ دالة (Procedure-function) هي دائمة لا إتجاهية (scalaire) .

القواعد :

1 ـ من الممكن كمُعامِل أن نستعمل أي نوع من المعطيات ونرسله إلى الدالة ، كما في الحالة SUBROUTINE .

2 ـ من الممكن أن لا يحتوي الأجراء ـ دالة على أي نوع من المتغيرات الوسيطة . في هذه الحالة ، إسم الأجراء ـ دالة يجب أن يكون مصرّحاً عنه في الفدرة التي تطلبه بواسطة الخاصية ENTRY . وفي أغلب الحالات ، إذا كان إسم الأجراء ـ دالة معروفاً من الفدرة (البلوك) الطالبة له (هذه هي الحالة إذا كان الأجراء ـ دالة هو من داخل الفدرة) فهذا التصريح هو غير ضروري .

3 ـ يجب تحديد ، للأجراء ـ دالة وللفدرة التي تطلبه ، خاصيات القيمة المُعادة لها بواسطة

الدالة . وفي غياب مواصفات كاملة ، ستأخذ لغة PL/ 1 الصيغ بالغلط ، مرتكزة على إسم الأجراء ـ دالة ، وحسب القواعد العادية .

نستعمل لذلك الصيغة RETURNS

يجب الإنتباه الى هذه القاعدة للخاصيات ، لأن الفدرة الطالبة لا تقوم بأي تحويل للقيمة المُرسلة لها بواسطة إجراء ـ دالة ، في مستوى إسم الدالة .

7.3 ـ إرسال المُعاملات

يُمكن أن تكون المُعامِلات والمتغيرات الوسيطية المناسبة من أي نوع من المعطيات إضافة إلى كونها من أي نوع من التنظيم :

ـ متحولات لا إتجاهية
ـ جداول
ـ تركيبات
ـ معطيات جبرية
ـ أو سمة
ـ سلاسل سمات أو بتات
ـ سجلات
ـ أسماء ادخال
ـ مؤشرات
ـ متحولات للحيزات
ـ متحولات برمجة متعددة .

8 ـ إدارة وتنظيم مباشر للذاكرة

لنشر إلى وجود أربع طرق لتخصيص الذاكرة في لغة PL/ 1 .

ـ ساكن (خاصية STATIC) .
ـ أوتوماتيكي (خاصية AUTOMATIC) .
ـ محكوم (خاصية CONTROLLED) .
ـ مؤشر (خاصية BASED) .

الخاصيات الأربع هي شاملة

في الحالتين الأخيرتين ، نتحكّم بالتخصيص بواسطة الأوامر ALLOCATE وFREE . في حالة التخصيص المحكومة ، الأوامر ALLOCATE المتتالية تُكدِّس حيزات الذاكرة (أقسام الذاكرة) ، ولكن الحيز الأخير هو المبلوغ . وهذا هو نفس الشيء في حالة التخصيص المؤشر ، مع الفارق الرئيسي أنه في أي لحظة من الممكن بلوغ جميع

الحيزات أو الأقسام من الذاكرة المكدسة .

8.1 ـ إضافة متحولة ومؤشر

عند التصريح عنها ، ترتبط المتحولة المؤشرة بواسطة متحولة خاصة تدعى مؤشر . يحتوي هذا المؤشر على عنوان . وبشكل عام هذا العنوان هو عنوان المتحولة المؤشرة موضع السؤال ، ولكن من الممكن تعديلها :

```
DECLARE X CHARACTER(150) BASED(P);
```

X هي متحولة (هنا : سلسلة سمات) وP هي عبارة عن مؤشر ، يحتوي على عنوان الذاكرة الفعلي للمتحولة X .

في الحالة المحدّدة أعلاه ، التصريح عن P كمؤشر هو مقرون ، من الممكن التصريح عن معرّف كمؤشر :

```
DECLARE R POINTER;
```

8.2 ـ مراجعة متحولة مؤشرة

مراجعة موقع متحولة مؤشرة يتم بواسطة رمز وصفي « يؤشر على » ، ويتألف من إشارة ناقص (-) وإشارة أكبر من)>) متتالية :

```
->
```

هذا الرمز ، الذي ليس إلا عبارة عن مؤثر ، جرى إختباره لدعوة لمتجه تأشير . واستعماله هو شبيه بالنقطة في كتابة الأسماء المفصلية للتركيبات .

مراجعة موقع المتحولة X المؤشرة بواسطة المؤشر المعتمد P يكتبُ على الشكل التالي :

```
P -> X
```

(X تدل على P)

من الممكن بشكل جيد إستعمال مؤشر مستقل معتمد ليوضع في القيمة الصحيحة :

```
R -> X.
```

1 ـ المتحولة المؤشرة لا يُمكن أن تحتوي على الخاصيات التالية :

```
EXTERNAL, VARYING, INITIAL.
```

2 ـ حدود الجداول المؤشرة وطول السلاسل المؤشـرة يجب أن تكون عبـارة عن ثوابت صحيحة عشرية .

3 ـ استعمال كلمة التحريك (Prefix d'activation) ((CHECK) variable : هو ممنوع في حالة المتحولات المؤشرة .

4 ـ التعليمات GET DATA وPUT DATA لا يُمكن أن ترسل متحولات مؤشرة .

5 ـ المؤشر الـوصفي ، الواضـح أو الضمني ، لا يُمكن أن يُؤشر أو يدُل عليـه . التعبير التالي ، حيث P, Q, R هي عبارة عن مؤشرات وX هي متحولة ، هو نظرياً ممكن في لغة PL/ 1 ، ولكنه غير مقبول حالياً من أي مُصرِّف :

```
P -> Q -> R -> X = ... ;
```

إذا TAB عبارة عن جدول مؤشرات ببعد واحد ، فالوصف التالي هو ممنوع :

```
PTAB(J) -> X
```

لـوصف X بواسـطة PTAB (J) ، يجب المرور بـواسطة مؤشر ملحق ، مثـلاً S ، وكتابة :

```
S = PTAB(J) ;   S -> X ... ;
```

6 ـ العمليات الوحيدة المسموحة على المؤشرات هي العمليات البولية المنطقية: = و =¬.

7 ـ تعليمة التخصيص هي غير مسموحة إلا من خلال مُحدِّد للمركز إلى مُحدِّد آخر .

8 ـ من غير الممكن إرسال مُركـز بواسطة إدخال / إخراج من نوع STREAM .

8.3 ـ إستعمال المؤشرات

يمكن لأربعة تعليمات أن تؤثر على المؤشرات :

```
SET = LOCATE ALLOCATE
```

SET ـ 8.3.1

تستعمل في الإدخال / الإخراج ، مع التعليمات READ أو LOCATE ، وفي التعليمة ALLOCATE .

عند القراءة ، تسمح الصيغة SET بتعريف متحولة مؤشرة مباشرة في حيز داريء للادخال / الاخراج (أو داريء Buffer) .

```
DECLARE MANIP FILE RECORD SEQUENTIAL,
        1 ALPHA BASED(Q),
               2 X ... ,
               2 Y ... ;
READ FILE(MANIP) SET(Q);
```

ستقرأ التسجيلة في حيز داريء والمؤشر Q سيحتوي على عنوان هذا الحيز . التركيبة

ALPHA سيتم تعريفها إذاً على الحيز داريء بسبب هذا الفعل ، وهذه الأخيرة تصبح مبلوغة مباشرة . ومن الممكن أيضاً التصريح عن متحولات أخرى مؤشرة بواسطة نفس المؤشر Q . جميع هذه المتحولات ستصبح إذاً معرّفة في نفس الوقت على حيز الداريء .

8.3.2 ـ التخصيص بواسطة السطر

مضمون المؤشر يُمكن أن ينقل إلى مؤشر آخر ، بلدون تحويل .

مثلاً ، إذا كان P وQ عبارة عن مؤشرات ، والمتحولة هي مؤشرة بواسطة P ، أو Q ، أو بواسطة مؤشر ثالث ، إذا قمنا بتنفيذ التعليمة التالية :

```
P = Q;
```

إذاً الكتابتين التاليتين : $P \rightarrow X$ و $Q \rightarrow X$ تقومان بمراجعة نفس موقع المتحولة المؤشرة .

LOCATE ـ 8.3.3

هذا الأمر هو أمر بالكتابة . وهو شبيه بأمر القراءة

```
READ ... SET.
```

مثلاً ، الأمر الكامل :

```
LOCATE X FILE(FMSORT) SET(P);
```

8.4 ـ الدوال التي تؤثر على المؤشرات

هناك إثنتان : NULL وADDR

ADDR ـ 8.4.1

تُرسل هذه الدالة العنوان الحقيقي لمتحولة مختلفة إلى المؤشر

```
DECLARE P POINTER;
P = ADDR(FOX);
```

NULL ـ 8.4.2

هذه الدالة هي بدون مُعامِلات . وهي ترسل إلى المؤشر قيمة ترفع كل معنى لهذه الأخيرة . هذا المؤشر يقال أنه مُصغَّر . هذه القيمة ليست سوى عنوان ممكن ، ومعروفة كذلك بواسطة اللغة .

8.5 ـ تحرير الذاكرة

الأمر FREE ، كما في حالة المتحولات المحكومة ، تُحرِّر حيّزاً مخصصاً لمتحولة مؤشرة .

```
FREE P->X, R->ALPHA, ... ;
```

وإذا لم نستعمل مؤشرات وصفية ، فسيأخذ المصرِّف بالغلط ، كما في الحالات الأخرى ، المؤشر المعتمد للمتحولة .

ملاحظة : يوجد أيضاً متحولات حيز (AREA) ، ومؤشرات نسبية في داخل هذه الحيزات (OFFSET) . ولن يجري طرحها هنا .

جميع هذه العناصر ، المؤشرات المطلقة والنسبية إضافة إلى متحولات الحيز ، تسمح بإجراء جميع العمليات الكلاسيكية للتسلسل النسبي أو معالجة اللوائح .

لنشر في النهاية إلى أن لغة PL / 1 تحتوي على سلسلة من الأجهزة المناسبة للبرمجة المتعددة ، تسمح بإنشاء أعمال مع مستويات أولوية محدَّدة ، وإنها قادرة على أن تتوسع ، بواسطة الماكروتعليمات (٪ تعليمات) ، هذه الأخيرة تنتمي إلى النحو العام للغة 1 / PL .

مراجع

مراجع أساسية

- الوثائق المناسبة للغة PL/ 1 ، والتابعة للمصرفات المستعملة .

من الممكن أيضاً إستشارة :

M. DORNBUSCH. *Le langage PL/1*. Dunod (1973).

Ch. BERTHET, *Le langage de programmation PL/1*, Dunod (Nouveau tirage 1982) (Ce chapitre est en fait un condensé de ce livre).

الفصل التاسع

بازيك BASIC

1 ـ مواضيع اللغة

1.1 ـ لعبة السمات

تستعمل لغة البازيك مجموعة من 58 سمة : الفراغ أو البياض ، يُستعمل كما في لغة فورتران لفصل مختلف عناصر التعليمة . لا تؤخذ الفراغات في الحسبان من قِبل المصنِّف ، ما إلاّ عنـدما تكون داخلة في سلسلة من السمات .

الأحرف ، أو السمات الأبجدية ، هي ال 26 حرفاً من الألفباء ،إضافة إلى السمات الثلاث .@ # $

ملاحظة : تُمثِّل الأدوات الطرفية الحرف 0 .

الأرقام من 0 إلى 9 :

السمات الخاصة هي التالية :

' هلال مزدوج

" هلالان مزدوجان

> أكبر من

= يعادل

< أصغر من

& ـو(تجارية)

) هلال يمين للاغلاق

(هلال يسار للفتح

+ زائد

− ناقص

* ضرب

/ قسمة

. نقطة

, فاصلة

؛ نقطة ـ فاصلة

: نقطتان

؟ علامة إستفهام

! علامة تعجب

1.2 ـ الثوابت

1.2.1 ـ الثوابت الجبرية

يُمكِن للثابتة الرقمية أن تكون حقيقية أو صحيحة ، إيجابية أو سلبية .

تتعلّـق القِيم الطرفية للثوابت والقيم الرقمية بالمصرِّفات وبالمكنات المستعملة (*) .

أمثلة على الثوابت الصحيحة :

```
    0
   12
  -32
12634
```

أمثلة على الثوابت الحقيقية :

```
  12.54
-  0.1234
123.1
   0.1345E+28
   0.3346E-17
```

في الحالتين الأخيرتين ، E + 28 و17 − E تعنيان على التوالي 10 بقوة 28 و10 بقوة − 17 .

1.2.2 ـ الثوابت من نوع سلاسل

الثابتة من نوع سلسلة (أبجعددية ، أو حرفية) هي عبارة عن سلسلة مختلفة بين هلال مزدوج (") . وإذا إستعملنا أهلة مزدوجة لتحديد السلسلة ، يُمكن عندها إستعمال الأبوستروف (') في داخل السلسلة ، والعكس هو صحيح . الطول الأقصى للسلسلة يتعلّـق بالمكنة(*) .

أمثلة :

(*) مثلاً ، بالنسبة للمكنة HP-85 ، وفي الشكل REAL ، تذهب الإعداد من $9.99...9 \times 10^{499}$ إلى -1.0×10^{-499} ومن 1.0×10^{-499} إلى $9.99...9 \times 10^{499}$.

```
"DEBUT DE PROGRAMME * 27"
"LISTE D'APPEL"
"RE_ENTREZ VOS DONNEES"
```

1.2.3 ـ ثوابت رمزية

تستعمل بعض فروع لغة بازيك ثوابت رمزية . وهي عبارة عن القيم π ، e ، $\sqrt{2}$. ويشار إليها بواسطة الأسماء &PI, &E و &SQR2 . وقيمتها هي :

```
&PI   : 3.14159...
&E    : 2.71828...
&SQR2 : 1.41421...
```

الدقة هنا ، تتعلق بالمكنة المستعملة .

1.3 ـ المتحولات

1.3.1 ـ قواعد الكتابة

تُمثّل المتحولة بواسطة حرف ، أو بواسطة حرف متبوع برقم .

تستعمل لغة بازيك صنفاً خاصاً من المتحولات من نوع سلسلة ، والمُمثّلة بواسطة حرف ، أو حرف ورقم ، متبوع بالسمة $. وتُدعى أيضاً متحولات حرفية .

أمثلة على المتحولات :

```
A  B1  B3  C0  *8  Z  X$  D$  B5$  R1$
```

يُمكن للمتحولة أن تكون لا إتجاهية أو بأبعاد .

1.3.2 ـ الجداول (الأبعاد)

الشكل العام لمتحولة بدليل هو :

$v(n, p)$

حيث v هو إسم المتحولة ، n, p عبارة عن دلائل . وهذه الأخيرة يمكن أن تكون عبارة عن ثوابت رقمية ، متحولات لا إتجاهية أو بدلائل ، أو تعابير جبرية مختلفة .

في الحالة التي نستعمل فيها تعابير جبرية ، تُحسب هذه الأخيرة وبعد ذلك تُقطع في قسمها الصحيح .

العدد الأقصى للدلائل في لغة بازيك هو 2 . وفي النهاية ، إسم المتحولة المدللة لا يُمكن أن يكون سوى حرفاً بسيطاً .

1.3.3 ـ التصريح عن المتحولات

يُصرَّح عن المتحولات اللاإتجاهية بشكل ضمني فقط . أما المتحولات ذات الأبعاد

(الجداول) فيصرَّح عنها إما بوضوح وإما ضمنياً بواسطة المراجعة الأولى لأحد العناصر .

وفي الحالة الأخيرة ، الأبعاد ستكون بالغلط (10) (متحولة بدليل واحد) أو (10, 10) متحولات بدليلين .

تبدأ الدلائل بالقيمة 0 .

التصريح الواضح للجدول يتم بواسطة التعليمة DIM ومن الممكن غالباً أن نجعل الدلائل تبدأ بالقيمة 1 بواسطة التعليمة :

```
OPTION BASE 1
```

ولكن جميع صيغ بازيك لا تسمح بهذه السهولة .

أمثلة :

```
10 DIM A(25)
20 DIM B(10,15)
30 DIM C(30), D(12,5), E(25,50)
```

تُحدَّد أبعاد الجداول دائماً بواسطة ثوابت صحيحة وإيجابية . يُمكن لاسم الجدول أن يكون شبيهاً باسم متحولة لا إتجاهية مستعملة في موقع آخر من البرنامج . ولكن جدولاً ببعد واحد لا يُمكن أن يكون بنفس إسم جدول آخر ببعدين .

عندما نصرِّح عن طول متحولة من نوع سلسلة من السمات ، نستعمل الأمر DIM ، كما بالنسبة للتصريح عن الجداول ، ولكن يجب أن نكتب طول السلسلة بين قوسين معقوفين .

مثلاً :

```
DIM A$ [30]
```

وعندما نرغب بالاحتفاظ بالمتحولات عند سلسلة البرنامج (متحولات مشتركة) ، نُصرِّح عنها بالتزامن في البرنامج الداعي وفي البرنامج المدعو ، بواسطة التعليمة COM .

مثلاً :

```
COM A1, A2, A3
COM B (20), C(30)
```

1.3.4 ـ إستعمال المتحولات

بشكل عام ، لا يوجد تخصيص ـ مسبق للصيغ للمتحولات اللاإتجاهية أو ذات الأبعاد في لغة بازيك ، على عكس ما نراه في لغة فورتران ، كوبول ، PL/ 1 . هكذا ،

يمكن أن تكون المتحولة الجبرية في البرنامج صحيحة أو حقيقية . واستعمال المتحولة في البرنامج هو الذي يُحدِّد طريقتها الحالية .

وفي النهاية ، جميع المتحولات الجبرية في لغة بازيك ، ذات الدليل ، أو اللاإتجاهية البسيطة ، هي مصفّرة قبل تنفيذ البرنامج .

الجداول ذات البعد ـ الواحد (المتجهات) أو ذات البعد الواحد (مصفوفات) يُمكن أن تستعمل بالعنصر ، أو تعالج في مجموعها ، أو عنصر بعد عنصر ، أو يُعاد تشكيل أبعادها بواسطة تعليمة MAT .

عندما يكون البازيك المستعمل موسعاً إلى متجولات من نوع سلاسل ، فأغلب العمليات (التخصيص ، المقارنة ، الإدخال ـ الاخراج ، التعليمة MAT) ، تنطبق على هذا النوع من المواضيع . في هذه الحالة ، يكون بتصرفنا دوال خاصة لمعالجة سلاسل السمات ، من نوع دليل ، طول ، إستخراج سلسلة ثانوية ، الخ . أسماؤها وتعريفها تتغيّر حسب البازيك الجاهز في السوق . ويجب على القارىء أن يراجع الوثائق المعتمدة لهذه الغاية .

1.4 ـ الالتحام

عملية الالتحام تؤدي إلى تعليق سلسلتين من السمات . يعادل طول السلسلة المنشأ مجموع أطول السلاسل الأولية .

مؤثر الإلتحام في لغة بازيك والسمة & (والتجارية) أمثلة :

```
A$ = "ABCDE"
B$ = "+++"
C$ = "12FGZ"
D$ = A$ & B$ & C$
```

مضمون D$ هو حالياً السلسلة "ABCDE+++12FGZ"

2 ـ كتابة التعليمات

تكتب كل تعليمة بلغة بازيك ، تنفيذية أو تصريحية على سطر واحد فقط .

وفي بعض الأحيان تسمح بعض أنواع البازيك بكتابة التعليمة على عدة أسطر، لتفادي الشكل المختزل للشاشات أو الطابعات . ولكن لا يوجد سوى سمة واحدة (CR) في نهاية التعليمة بازيك .

تحتوي كل تعليمة على رقم ، في بداية السطر . تُنفَّذ التعليمات على التوالي ، حسب نظام ترتيبها . وفي النهاية ، يمكن لأرقام التعليمات أن يكون لها زيادات مختلفة : نكتبها بشكل عام 10 بعد 10 . لكل رقم سطر قيمة وسم معينة (Label) .

مثلاً :

```
10 LET A=1
20 A=A+1
30 B=30
40 C=A+B
50 PRINT C
60 GO TO 20
99 END
```

البرنامج بلغة بازيك هو عبارة عن سلسلة منتظمة من التعليمات . يُمكن خلط التعليمات المنفذة وغير القابلة للتنفيذ فيها بينها . إنتقال التحكم إلى تعليمة تصريح يؤدي إلى تفريع الى التعليمة التالية مباشرة . يحتوي السطر الأخير من البرنامج على تعليمة END ، التي تنهي فيزيائياً ومنطقياً البرنامج . وفي أغلب الحالات ، من الممكن وقف البرنامج منطقياً في أي مكان ، بواسطة تعليمة STOP ، التي تعادل تفريعاً لى التعليمة END .

التعليمة REM (ملاحظة) ، تسمح بإدخال النصوص والملاحظات الى البرنامج .

مثلاً :

```
150 REM ROUTINE D'ANALYSE.
```

تُهمل الفراغات بواسطة بازيك ، ما عدا في REM أو في ثابتة من نوع سلسلة من السمات ما عدا بعد أول حرف من كلمة ـ مفتاح .

3 ـ تعليمة التبادل

تتمثل تعليمة التبادل في لغة بازيك بواسطة السمة =. ومن الممكن إضافة الكلمة LET الى بداية التعليمة ، وهي إختيارية دائماً .

الشكل العام :

LET $v_1, v_2, \ldots, v_n = x$

حيث $v_1, v_2, \ldots, v_n$ هي متحولات وx هي ثابتة ، متحولة أو تعبير . يتم التبادل ، كما في اللغات الكلاسيكية ، من اليمين لجهة اليسار . وفي النهاية ، يُمكن لعدة تعليمات نقل أن تكون موجودة على نفس السطر . وهي تفصل إذاً بواسطة فواصل ، والكلمة LET إذا إستعملت ، لا يجب أن تتواجد سوى مرة واحدة في بداية السطر .

أمثلة :

```
10 LET A=12.5
20 LET X(K, J)=A+1
30 LET C, D, E=10, F=5.3, G=K
40 X1, X2=A, H1=1.5, Z=1000
```

لنشر إذاً إلى إن صيغة المتحولة تتعلق باستعمال هذه الأخيرة ، لا يوجد مفهوم تحويل للصيغة من يمين إلى يسار الإشارة = في لغة بازيك .

4 ـ التعابير الجبرية

4.1 ـ المؤثرات الجبرية

المؤثرات الجبرية هي التالية :

+ الجمع

− الطرح

* الضرب

/ القسمة

** بقوة (أس) .

أمثلة :

مجموع X وY	X + Y
الفرق K ناقص 1	K − 1
ضرب A بواسطة 3.5	A*3.5
قسمة G على 5.1	G / 5.1
المتحولة X1 مرفوعة إلى القوة K	X1**K

4.2 ـ الدوال الجوهرية

وهي متكاملة في اللغة . ولا يوجد لائحة نهائية بها ، ولكل مصرِّف مجموعته الخاصة من هذه الدوال .

SIN(X)	
COS(X)	
TAN(X)	
COT(X)	
ASN(X)	
ACS(X)	
HSN(X)	
HCS(X)	
HTN(X)	
EXP(X)	دالة أسية
LOG(X)	لوغاريتم طبيعي
LGT(X)	لوغاريتم بقاعدة 10
SQR(X)	جذر تربيعي
RND(X)	مولّد أعداد شبه ـ عشوائية
INT(X)	قسم صحيح من X
SGN(X)	إشارة X (1 − = − 1 ، 0 أو 1 + حسب الحالة)
ABS(X)	القيمة المطلقة لـ X

أمثلة على الاستعمال .

```
10 LET A=SQR(12.5)
20 B=EXP(1+X)
30 C=LOG(Y+2.5)+LOG(Z-3)
```

4.3 ـ الدول المُنشأة بواسطة المستعمل

من الممكن تعريف دوال جديدة في برنامج بواسطة التعليمة DEF . هذه الدالة لن تكون معروفة سوى في البرنامج المعرَّفة فيه . ويُستدل عليها بـواسطة حـرف ، مسبوق مباشرة من الحرفين FN .

أمثلة :

```
 10 DEF FND(X)=1+X+X×X
 20 DEF FNA(X)=LOG(1+X)+3.5*X
    .
    .
110 LET A=FND(B-4.2)
120 C1=FNA(D1)-FNA(D2-15.7)
```

الخ .

5 ـ تعابير منطقية

5.1 ـ مؤثرات منطقية

وهي عبارة عن مؤثرات العلاقة

< أو LT أقل من

> أو GT أعلى من

= أو EQ يعادل

<= أو LE أقل أو يعادل

>= أو GE أكبر أو يعادل

<> أو NE يختلف عن ..

5.2 ـ تعابير منطقية

التعبير المنطقي يقارن بين تعبيـرين جبريـين . ويُمكن أن يأخـذ قيمتين : حقيقـة وغلط .

الشكل العام هو :

حيث x ، y هما عبارة عن تعبيرين جبريين (=ثوابت، متحولات أو تعابير) و op هو أحد المؤثرات الستة للعلاقات .

أمثلة :

```
A<B
X+Y-1  EQ  0
Z*3.5  <=  SQR(1+X)
```

لا يوجد أبداً متحولات منطقية ، ولا تعابير منطقية عامة في لغة بازيك ، على عكس فورتران ، كوبول ، PL / 1 . التعابير المنطقية في بازيك تستعمل في إطار التعليمة IF (أنظر لاحقاً) . وفي أغلب الحالات ، تحتوي الصيغ الموسعة للغة بازيك على أغلب المؤثرات المنطقية (NOT ، AND ، OR مع رموزها) إضافة إلى إمكانية كتابة التعابير المنطقية العامة .

5.3 ـ حسابة التعابير أولوية المؤثرات

قواعد حسابة التعابير في لغة BASIC هي نفسها القواعد المستعملة في جميع اللغات الكلاسيكية . ويُمكن للقارىء أن يراجع بشأنها الفقرة 5.3 في الفصل فورتران .

6 ـ التحكُّم بدوران البرنامج

6.1 ـ تفريع غير مشروط

6.1.1 ـ الأمر GOTO البسيط

الشكل العام هو :

```
GO TO n
```

حيث n هي رقم السطر (= وسم) .

مثلاً :

```
110 GO TO 230
```

6.1.2 ـ الأمر GO TO المحسوب

الشكل العام هو :

$\underline{GO\ TO}\ n_1, n_2, \ldots, n_p\ \underline{ON}\ x$

حيث $n_1, n_2, \ldots, n_p$ هي عبارة عن أرقام الأسطر (= وسم) وX هي عبارة عن تعبير جبري . التفريع سيتم إلى السطر n_i إذا كان القسم الصحيح من x يعادل i . وإذا كان العدد i ليس موجوداً بين 1 وp فسيُهمل الأمر .

مثلاً :

```
240 GO TO 15, 20, 260, 310 ON K
```

6.2 ـ التفريع المشروط

الشكل العام :

```
IF x op y THEN n1 ELSE n2
```

y, x هي عبارة عن تعابير جبرية (ثوابت ، متحولات ، معادلات) ، op هو مؤثر علاقة .

n_1, n_2 هي عبارة على أرقام الأسطر (= أوسمة) .

إذا كان التعبير المنطقي حقيقياً ، فالتحكم يعبر إلى السطر n . وفي الحالة المعاكسة ، يمر إلى n_2 .

القسم ELSE n_2 هو إختياري . وإذا أهملت ، وكان التعبير المنطقي غلط ، فالتحكم يعبر إلى السطر التالي :

أمثلة :

```
110 IF A>B THEN 250
210 IF X=0 THEN 300 ELSE 10
```

6.3 ـ حلقات البرنامج

الأمر FOR .. NEXT

الشكل العام هو :

```
FOR v = x TO y STEP z
 .
 .
NEXT v
```

حيث v هي عبارة عن متحولة لا إتجاهية (Scalar) و z, y, x هي تعابير جبرية .

تدور الحلقة لجميع القيم v الموجودة بين القيمة x والقيمة y بخطوة تعادل z .

القسم STEP z هو إختياري . وإذا جرى إهماله ، تُعتبر z = + 1 .

القواعد :

- من الممكن تداخل الحلقات .
- من الممكن الخروج من الحلقة في أي موقع فيها .
- يمنع تعديل قيمة الدليل v في داخل الحلقة .
- يمنع تعديل التعابير والمتغيرات الوسيطة x, y, z بداخل الحلقة .
- من الممكن التفريع من الخارج إلى داخل الحلقة (ولكن لا يُنصح بذلك) .
- يمكن لأمر واحد NEXT من إنهاء عدة حلقات .

أمثلة :

```
100 FOR I=1 TO 15
110 LET S=S+I
120 NEXT I
200 FOR X=0.5 TO 22 STEP 0.1
210 Y=SQR(X*X+1.2)
220 PRINT X, Y
230 NEXT X
300 FOR K=10 TO 100 STEP 10
310 FOR N=1 TO 10
320 LET A(K, N)=K*N-1
330 NEXT N, K
```

6.4 ـ البرامج ـ الثانوية

6.4.1 ـ الأمر GOSUB البسيط

الشكل العام :

GOSUB n

حيث n هو رقم التعليمة (= رقم السطر) .

يُنقل التحكم إلى التعليمة n . في نفس الوقت يوجد خزن لعنوان العودة ، هو عنوان العنوان التالي مباشرة للأمر GOSUB .

مثلاً :

```
120 GOSUB 350
```

6.4.2 ـ الأمر GOSUB المحسوب

الشكل العام :

GOSUB $n_1, n_2, \ldots, n_p$ ON x

حيث n₁، n₂، ....، n هي أرقام الأسطر ، وx هو عبارة عن تعبير أو معادلة جبرية . طريقة العمل هي نفسها كطريقة عمل الأمر GOTO المحسوب (أنظر الفقرة 6.1.2) . الفرق مع هذه التعليمة الأخيرة هو في وجود خزن لعنوان العودة .

مثلاً :

```
150 GOSUB 100, 300, 210, 540 ON M
```

6.4.3 ـ الأمر RETURN

هذا الأمر يُعيد التحكم إلى عنوان العودة المُخزَّن بواسطة التعليمة GOSUB .

مثلاً :

```
120 GOSUB 350
.
350 Z=SQR(X*Y+3.5)
360 PRINT X, Y, Z
370 RETURN
```

7 ـ إدخال ـ إخراج

7.1 ـ العمليات الداخلية

7.1.1 ـ التعليمة DATA

الشكل العام هو :

DATA $c_1, c_2, \ldots, c_n$

حيث $c_1, c_2, \ldots, c_n$ هي عبارة عن ثوابت جبرية . ويُمكن أن تكون أيضاً عبارة عن ثوابت حرفية .

مثلاً :

```
150 DATA 25, 351.5, 1, 0.5, 12,
                        -35.1,2
```

يتشكّـل جدول المعطيات عند التصريف . وفي الأساس ، يُشكّـل المصرِّف جدولين محدّدين : واحد للمعطيات الرقمية ، وآخر للثوابت الحرفية .

7.1.2 ـ الأمر READ

الشكل العام :

READ $v_1, v_2, \ldots, v_n$

حيث $v_1, v_2, \ldots, v_n$ هي عبارة عن متحولات .

تُقرأ القيم n من الجدول DATA ، وتُنقل إلى المتحولات $v_1, v_2, \ldots, v_n$.

مثلاً :

```
200 READ A, B, C
```

7.1.3 ـ الأمر RESTORE

مؤشر القراءة يُعاد تركيزه على أول تسجيلة من الجدول DATA .

7.1.4 ـ التعليمة ON EOD

الشكل العام :

ON EOD GO TO n

n هي رقم السطر (= الوسم) .
في حالة نهاية المعطيات (End Of Data = EOD) في لائحة DATA ، سيجري تنفيذ الأمر GO TOn .

إذا جرى إهمال GO TO ، فسيتم إرجاع الشرط النموذجي للنظام .

7.2 ـ العمليات الخارجية

7.2.1 ـ الأمر INPUT

```
INPUT  v1, v2, ..., vp
```

$v_1, v_2, \ldots v_p$ هي عبارة عن أسماء متحولات .

يجب إدخال قِيم المتحولات بواسطة الأداة الطرفية أو لوحة الملامس .

وبشكل عام ، يطبع البازيك علامة إستفهام على شاشة الأداة الطرفية عند تنفيذ التعليمة INPUT ليشير بذلك إلى الرغبة بإدخال المعطيات المناسبة للمتحولات الواردة بعد الأمر INPUT .

عند الإدخال ، تُفصل المعطيات بواسطة فواصل . نهاية العملية يُشار إليها باستنفاذ اللائحة ، وإما بإدخال السمة / (القسمة) .

مثلاً :

```
130 INPUT A, B, C, X
```

7.2.2 ـ الأمر PRINT

```
PRINT  x1, x2, ..., x
```

حيث $x_1, x_2, \ldots, x_p$ هي ثوابت ، متحولات أو معادلات . يتم طبع القِيم x_i إما بنسق مُفصَّل (فصل بواسطة فواصل ، كما في المثل 220) وإما بنسق مضغوط (فصل بواسطة نقاط ـ فاصلة ، مثلاً 350) .

أمثلة :

```
220 PRINT A, B, C, D
350 PRINT X; Y; Z; T
```

ملاحظة : من الممكن التحكم ببعض طرق طباعة النتائج بواسطة الأمر PRINT البسيط ، الفواصل ، النقاط ـ الفواصل . وينصح القارىء بمراجعة الوثائق الخاصة بالاستعمال، والإمكانيات المقدمة والمتعلقة بالمصمِّم .

7.2.3 ـ التعليمة IMAGE (أو FORMAT(*))

تستعمل هذه التعليمة مع الأمر PRINT USING
والشكل العام هو :

$: c_1\ s_1\ c_2\ s_2\ c_3\ s_3\ \ldots\ c_n\ s_n$

حيث $c_1, c_2, \ldots, c_n$ هي عبارة عن سلاسل من السمات ، يُمكن أن تحتوي على أي سمة غير # (وقد تهمل) .

$s_1, s_2, \ldots, s_n$ هي عبارة عن مواصفات النسق . النسق S الصحيح ، أمثلة :

(4 أرقام) `# # # #`

(إشارة وثلاثة أرقام) `- # # #`

(إشارة ورقمين) `+ # #`

النسق s الحقيقي

(5 أرقام ، 3 منها كسرية) `##.###`

(1 إشارة ، 3 أرقام ، 2 بعد الفاصلة) `- #.##`

النسق s في الشكل E ، أمثلة :

```
  #.##!!!!
- ##.##!!!!
```

علامات التعجب !!!! تدل على مميزات بقوة 10 ، مسبوقة بـ E وباشارة :

أمثلة :

```
230:  "VALEUR="  ##.###
240:  ###  #.##  #.#  #.#
```

7.2.4 ـ الأمر PRINT USING

الشكل العام :

PRINT USING $n; x_1, x_2, \ldots, x_p$

حيث n هو رقم السطر المناسب للتعليمة IMAGE ، و $X_1, X_2, \ldots X_n$ هي تعابير (متحولات ، ثوابت معادلات) .

أمثلة :

```
100 PRINT USING 230; X1
150 PRINT USING 240; A, B, C1, C2
```

(*) هذه التعليمة يُمكن ان تكون بأشكال مختلفة حسب نوع البازيك المستعمل .

Converted by Tiff Combine - (no stamps are applied by registered version)

7.2.5 ـ التعليمة ON EOF

هذه التعليمة تستعمل مثل ON EOD (فقرة 7.1.4) ولكن للعمليات الخارجية .

7.3 ـ العمليات على السجلات

من الممكن في بعض أنواع بازيك ، إجراء عمليات إدخال ـ إخراج على سجلات خارجية يُصرَّح عن السجلات إذاً بواسطة تعليمة FILE . ويمكن أن تكون ببلوغ متوالي أو عشوائي . كما ويُمكن أن تكون أيضاً في صيغة ASCII أو ثنائية . يوجد سجلات ASCII متتالية ومباشرة .

الأوامر المستعملة هي بشكل عام READ وWRITE ، مع مراجعة أرقام السجلات المنطقية ، وغالباً GET وPUT ، أو INPUT وOUTPUT .

لا يوجد أي معايرة في هذا الحقل ، والفروقات بين صِيغ البازيك الموجودة في الأسواق هي كبيرة ، ويجب على القارىء أن يرجع بانتظام إلى الوثائق الخاصة بمكنته وخاصة لجهة إمكانيات الإدخال ـ الإخراج .

8 ـ العمليات على الجداول

كل جدول ببعد واحد (متجه) أو ببعدين (مصفوفة) ، ومُصرَّح عنه ضمنياً أو بشكل جلي (التعليمة DIM) ، يُمكن أن يُعالج ويدخل في تعليمات خاصة بالمصفوفات من نوع MAT .

هذه التعليمة تدل على الجدول في مجموعه . ويُمكن أن تكون عملية تخصيص أو تعليمة جبرية ، أو إدخال ـ إخراج ، الخ .

إضافة لذلك ، فكل تعليمة MAT (ما عدا MAT PRINT وMAT PRINT USING) يُمكن أن تعيد تشكيل أبعاد الجدول المعتمد ، بشرط أن لا يزيد هذا التشكيل للابعاد عن العدد الكامل للعناصر في الجدول ، ولا يُغيِّر الأبعاد .

أمثلة على الاستعمال في إعادة تشكيل الأبعاد :

```
10  DIM(30,30), Y(100)
    .
    .
110 MAT X=(12)
120 MAT Y(50)=(LOG(A+B))
    .
210 MAT X(10,20)=(10)
```

التعليمة 10 تصرِّح عن الجداول x وy . التعليمة 110 تُخصِّص القيمة اللاإتجاهية 12

إلى جميع العناصر في x ، وهي 900 عنصراً في هذه اللحظة . التعليمة 120 تعيد تشكيل الجدول ببعد واحد إلى 50 عنصراً وتخصص القيمة اللااتجاهية المحسوبة إلى كل عنصر من هذه العناصر . وفي النهاية ، التعليمة 210 تعيد تشكيل الجدول ببعدين X إلى 10 « أسطر » و«10» أعمدة ، وتخصِّص إلى كل من هذه الـ 200 عنصر القيمة اللااتجاهية 10 .

التخصيص :

$\underline{MAT}\ m_1 = m_2$

- m_1, m_2 هي عبارة عن جدولين متشابهي الأبعاد .
- m_1 يمكن أن يُعادل تشكيل أبعاده . في هذه الحالة ، يكفي وضع القيم (القيمة) الجديدة للأبعاد بين أهلّة مباشرة بعد m_1 .

  هذه القيم يُمكن أن تكون عبارة عن ثوابت ، متحولات أو تعابير .
- كل عنصر من m_2 يمكن أن يستبدل العنصر المناسب من m_1 .

الجمع ، الطرح

$\underline{MAT}\ m_1 = m_2 + m_3$
$\underline{MAT}\ m_1 = m_2 - m_3$

- يُمكن أن يكون m_1 بأبعاد .
- m_1, m_2, m_3 يجب أن تكون شبيهة بالأبعاد .
- التعليمة تقوم بإجراء عملية الجمع (الطرح) لعنصر بعد عنصر من m_2 وm_3 .

الضرب .

$\underline{MAT}\ m_1 = m_2 * m_3$

- m_1 يُمكن أن يعاود تشكيل أبعاده .
- العملية تجري عملية الضرب المصفوفي لـ m_2 بـ m_3 ، مع انتقال للعناصر المناسبة الى m_1 . أبعاد المصفوفات m_1, m_2, m_3 يجب أن تكون متوافقة : إذ $m_2(n,p)$ و$m_3(p,m)$ ، $m_1(n, m)$.

العكس

$\underline{MAT}\ m_1 = \ INV(m_2)\ ,\ d$

- m_1 يمكن أن نعيد تشكيلها .
- n_2 هي بأبعاد متوافقة .

● d هي متحولة، إختيارية ، مخصصة لاستقبال قيمة المُحدِّد في m2 .

تخصيص قيمة ثابتة

$\underline{\text{MAT}}\ m_1 = (x)$

● m1 يمكن أن نعيد تشكيلها .

x هي ثابتة ، متحولة أو تعبير لا إتجاهي .

قراءة داخلية

$\underline{\text{MAT}}\ \underline{\text{READ}}\ m_1, m_2, \ldots, m_n$

● m1, m2 ... m n يُمكن أن يعاود تشكيل أبعادها .

القراءة تتم سطراً بعد سطر في جدول بتعليمة DATA .

القراءة الخارجية

$\underline{\text{MAT}}\ \underline{\text{INPUT}}\ m_1, m_2, \ldots, m_n$

● m1, m2, ...,mn يُمكن أن يُعاود تشكيلها .

$\underline{\text{MAT}}\ \underline{\text{PRINT}}\ m_1, m_2, \ldots m_n$

m1, m2, ..., mn لا يُمكن أن يُعاود تشكيلها .

● تتم القراءة حسب القواعد المناسبة للتعليمة PRINT .

$\underline{\text{MAT}}\ \underline{\text{PRINT}}\ \underline{\text{USING}}\ n, m_1, m_2, \ldots, m_n$

● الكتابة تتم حسب القواعد المناسبة للتعليمة PRINT USING .

● n هي رقم تعليمة الصورة (أو النسق) .

تعليمة أخرى MAT

TRN : للتبديل

CON : مصفوفة وحدة

DET : حسابة المُحدِّد

ZER : مصفوفة صفر

IDN : مصفوفة تعريف

التعليمات MAT هي مفيدة ، ولكن يبقى على القارىء والمبرمج أن يُحكم على فائدة إستعمالها ، وفي بعض الأحيان كتابة الخوارزم الخاص به .

مراجع

مراجع أساسية

الوثائق التقنية التابعة للغة بازيك والخاصة بالمكنة .

ومن الممكن أيضاً إستشارة الكتب التالية :

J. ARSAC, *Premières leçons de programmation*, Cedic-Nathan, Paris, 1980. (L'ouvrage, déjà cité, comporte également la transcription des exemples de programmes en LSE et en PASCAL).

On pourra également se reporter à :

A. CHECROUN, *BASIC, programmation des microordinateurs*, Dunod, Réédition 1982.

الفصل العاشر

المعلوماتية البعيدة

1 ـ تعريف

نقول بوجود معلوماتية بعيدة ، عندما يكون هناك تعاون كامل بين وسائط المعلوماتية ووسائط الاتصالات البعيدة .

وبالتحديد ، فلنفترض نظامين للمعلوماتية A وB (شكل 1) .

A وB يمكن أن يكونا حاسبات ، أدوات طرفية ، أدوات محيطية خاصة ، الخ .

مثلاً ، في الحالة الأكثر استعمالاً ، A هو حاسب وB هو أداة طرفية ، من نوع Teletype .



شكل 1 ـ وصلة معلوماتية بعيدة

D هو طول الوصلة الفيزيائية بين A وB . وفي أغلب الحالات ، هذه الوصلة هي مادية بواسطة خيط موصل بواحد أو عدة أزواج . هناك معلوماتية بعيدة ، إذا كان الطول D كبيراً ، بدون أن يكون حاجة لأي تغيير في A وB . وبشكل آخر ، الوصلة «D» هي وصلة معلوماتية بعيدة إذا كانت القيمة «D» مختلفة .

من الممكن إنشاء شبكة معلوماتية بعيدة حقيقية على طاولة ، بطول D يعادل بحوالى عدة عشرات من السنتمرات ، وعلى العكس بإمكاننا ربط أداة طرفية إلى حاسب من مسافة 50 متراً دون أن يكون هناك أية معلوماتية بعيدة .

عملية جعل المسافة D مختلفة تفترض إستعمال بعض التقنيات الخاصة ، التي هي موضوع هذا الفصل .

2 ـ تقنيات الإرسال

2.1 ـ تكويد المعطيات

صيغة تكويد المعطيات لارسالها لا يختلف أبداً ، وأساساً ، عما هو جاري في حدود الحاسبات .

نحتاج إلى أربعة أرقام ثنائية لتمثيل المعلومات الرقمية . ولتكويد الأحرف نحتاج الى رقمين إضافيين . نظام التكويد الذي يُزود بمجموعة كاملة من الأرقام والأحرف هو الكود BCD (Binary coded Decimal) أو DCB (Decimal Coded Binary) مع 36 سمة مختلفة (الملحق) .

هذا النظام هو كاف لإرسال المعطيات ، التي نحتاجها ، والإشارات ، وكما في كل حاسب ، إضافة إلى عدد غير قليل لسمات التحكّم . تحكم سمات التحكم إرسال المعطيات ، وتفصل الرسائل ، وتؤمن عمليات الخدمة : تحريك النظام ، إلغاء الطباعة ، الخ .

النظام (Extended Binary Coded Decimal Interchange Code) ، أي العشري الموسع المكود ثنائياً) هو كما يدل إسمه توسيع للكود BCD . ويستعمل 8 أرقام ثنائية بدلاً من 6 . وبالتالي هناك 256 مجموعة ممكنة ، منها 109 مستعملة فعلياً . يُستعمل هذا الكود غالباً لإرسال سمات الحاسب بالبايتة ، بدون أن يكون هناك حاجة إلى إجراء أي تحويل بين المكنة والخط . جدول التكويد EBCDIC هو موجود في الملحق .

ظهرت الأبجدية الدولية n°5 (IA5) نتجت جهود في المعايرة من قبل اللجان : CCITT (Comité consultatif International du telegraphe et du téléphone) ، واللجنة ISO (International Standards Organization) ومجموعات دولية أخرى . ولقد جرى تطويرها للاجابة على حاجات الكود النموذجي التلغرافي . أما الأبجدية الجديدة فنتجت عن عرض للجنة Americain standards Association (ASA) : التي أصبحت بعد ذلك ANSA (Americain National Standards Association) ، المعروفة بالاسم Americain Standard Code for Information Interchange (ASCII) . ولا تُعرف الآن إلا بهذا الإسم .

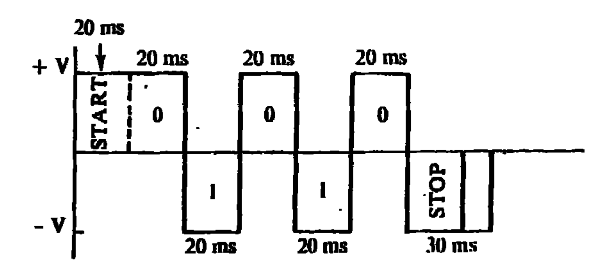
ولقد جرى قبول الألفباء ASCII نهائياً من قبل اللجنة CCITT في أوكتوبر سنة 1968 . جدول ASCII هو موجود في الملحق .

2.2 ـ سرعة الإرسال

2.2.1 ـ سرعة التضمين

يجب ألا نخلط بين السرعة الثنائية (سرعة الإرسال) مع سرعة التضمين .

سرعة التضمين هي عكس مدة الإشارة البسيطة (شكل 2) . وتقاس في البود (Baud) .



شكل 2 ـ تضمين بسيط بمستويات (إشارة إطلاق ـ وقف)

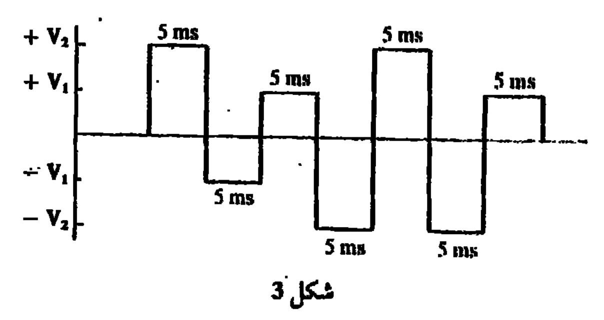
مثلاً ، إذا كانت سرعة الإشارة البسيطة هي عبارة عن 20 مللثانية ، فسرعة التضمين هي حوالي 50 بود ، سرعة التضمين القصوى المقبولة بواسطة الخط هي كمية مرتبطة مباشرة بنطاق الترددات العابرة لهذا الخط . لهذا يوجد خطوط قادرة على نقل وإحتمال سرعات تضمين من 50 بود (مثلاً Telex) ، من 1.200 ، 2.400 ، 4800 (التلفون) ، وحتى 84 كيلوبود وأكثر .

2.2.2 ـ سرعة الإرسال

تقاس سرعة الإرسال بعدد البتات في الثانية . وهي تحسب كمية المعلومات المرسلة في وحدة زمنية . وفي إرسال تسلسلي ، وتحدد في :

$$Vtr = \frac{1}{T} \log_2 n$$

حيث T هي مدة الإشارة البسيطة وn هي عدد الحالات التمثيلية لهذه الإشارة البسيطة .



شكل 3

مثلاً ، في الشكل 3 ، مدة كل إشارة بسيطة ، T ، هي 5 ms . يمكن لكل إشارة

بسيطة أن تأخذ الحالات $+V_1$، $+V_2$، $-V_1$ و $-V_2$. معنا إذاً $n = 4$.

بواسطة تطبيق الصيغة (1) ، نجد إن سرعة الإرسال Vtr هي 400 بتة / ثانية ، بينما سرعة التضمين هي 200 بود .

في الحالة السابقة (شكل 2) ، كان يُمكن للاشارة البسيطة أن تأخذ الحالات $+V$ و $-V$. سرعة الإرسال تتطابق إذاً مع سرعة التضمين ، لأن $\log_2 2 = 1$.

وهذا ما يجري في بعض الحالات ، ونحن لا نرتكب أي خطأ إذا طابقنا الحالتين معاً . ولكن من المهم في بعض الاحيان ان نعرف الفرق بين التعريفين .

2.2.3 ـ الخطوط الجاهزة

ـ شبكة تلفونية

الشبكة التلفونية هي مستعملة في الإرسال حتى 200 بود ، الإشارة البسيطة يُمكن أن تأخذ الحالتين 48V + و48V - . وهذا يسمح بالتشغيل full-duplex ، والإرسال يمكن أن يكون مختلفاً عن الإستقبال (full-duplex يعني إزدواجية كاملة أي بالاتجاهين) .

في فرنسا هناك مراكز الخدمات التالية :

- شبكة تلغرافية تبديلية (تلكس CCITT V 10) ، تسمح بتوصيلات إلى 50 بود ، كود CCITT رقم 2 إلى 5 لحظات .
- تلكس 200 بود ، على شبكة تبديلية .
- خطوط مُخصَّصة ثابتة حتى 200بود .
- شبكة تلفونية .
- الشبكة المبدلة العادية بشريطين ، تسمح بالعمل حتى 200 بود في الصيغة duplex (CCITT V23) ؛
- الوصلات الخاصة ، المخصصة دائماً للمستعمل والموزعة :
- خط بشريطين بنوعية عادية ، وبإمكانية شبيهة بالشبكة التبادلية .
- خطوط من أربعة أشرطة بنوعية عادية ، للشبكة المتعددة المراكز ، full-duplex ، الخ ، حتى 2400 بود .
- خطوط من 4 أشرطة بنوعية جيدة (M.102) ، للارسال السريع ، حتى 9600 بود .
- الشبكة DADUCEE ، شبكة تبديلية بخطوط من 4 أشرطة وبنوعية جيدة (M.102) ، والتي تسمح بإجراء توصيل حتى 72000 بود وفي داخل مناطق من المدينة ، وحتى 4800 بود إلى مسافة بعيدة .

نعني بسرعة الإرسال في بعض الأحيان ، عدد السمات في الثانية : أي السرعة في البود مقسومة على 10

وصلات بنطاق واسع

يتعلّق ذلك بالكابلات المحورية أو الحزمات الهرتزية ، التي تسمح بسرعة إرسال من 48 كيلو بود إلى أكثر من 1 مليون بود .

خطوط الإرسال في نطاق أساسي

هي عبارة عن الخطوط بنطاق واسع للعبور (بضع عشرات كيلوهرتز) . وتستعمل بواسطة موديم خاص .

2.3 ـ طرق الإرسال

الإشارة المستطيلة لا يُمكن نقلها كما هي بسرعة كبيرة إلى مسافة بعيدة : يجب إذاً إجراء تحويل عليها . هكذا ، نستخدم هذه الإشارة لتضمين موجة جيبية بترددة معينة ، وهذه الموجة المضمّمة هي التي نرغب بإرسالها بواسطة شريط أو بواسطة الراديو .

2.3.1 ـ التضمين السعوي

تُضمّـن الإشارة سعة الترددة المرسلة (شكل 4) .

وإذا اعتبرنا إن f هي الترددة الحاملة ، s هي تلك الخاصة بالإشارة (مع الإفتراض بأن الإشارة هي جيبية) ، فالموجة المُضمّمة تتألف عملياً من ثلاث ترددات :

f الترددة الأصلية

$f + s$ المجموعة الحاملة والمُضمِّنة

$f - s$ الفرق بين الحاملة والمُضمِّنة .

من الكلاسيكي ألا يجري إرسال سوى f-s فعلياً ، بهدف زيادة سرعة الخطوط . وعملياً ، إذا كانت f ثابتة فالترددة s تحتمل تغييرات مختلفة ، ومن الممكن النظر إلى f- s و f + s كنطاق للترددات . لهذا نتكلم عن شريط أو نطاق جانبي f-s . ولن يقوم نظام المصفّيات (filter) بالسماح إلا لهذا الشريط بالعبور . في الطرف الآخر من الخط ، نقوم بكشف الإشارة ، بإضافة وطرح إلى f ، المولودة مركزياً ، الإشارة f-s : فنحصل عند ذلك على (شكل 5) :

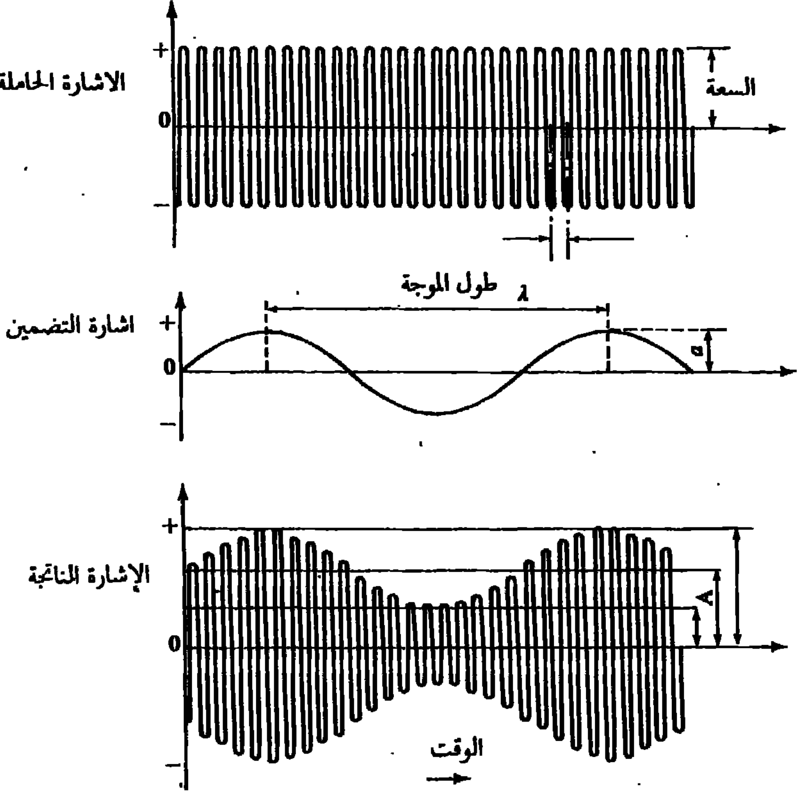
f

الذي نلغيه بالتصفية $f + (f - s) = 2f - s$

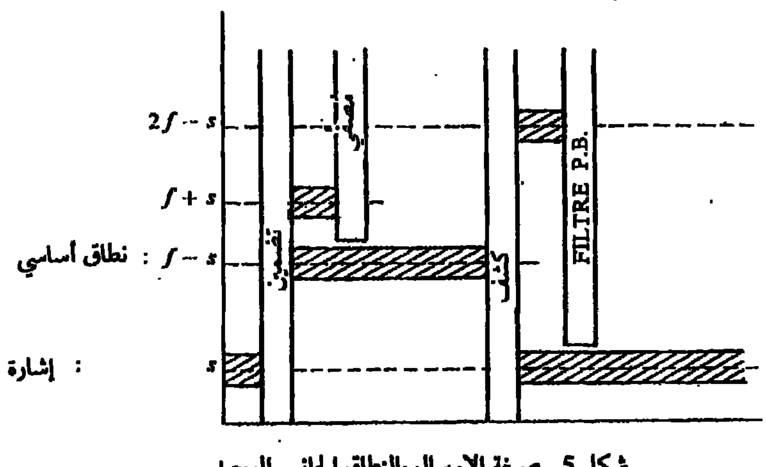
الإشارة الأصلية . $f - (f - s) = s$

الأداة التي تقوم بالتضمين والكشف تدعى modem (modulateur- demodulateur) وفي العربية مضمن - كاشف أو مضشف .





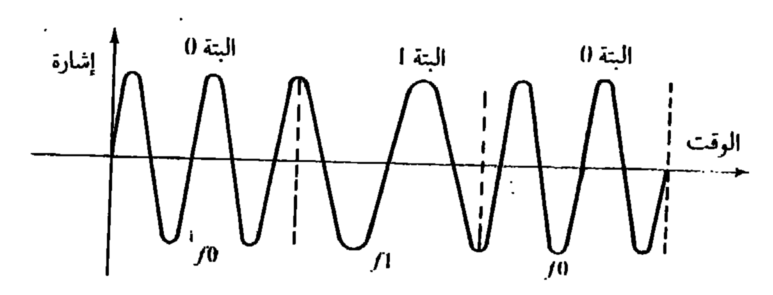
شكل 4 ـ صيغة التضمين السعوي



شكل 5 ـ صيغة الإرسال بالنطاق الجانبي الوحيد

2.3.2 ـ التضمين بالترددة

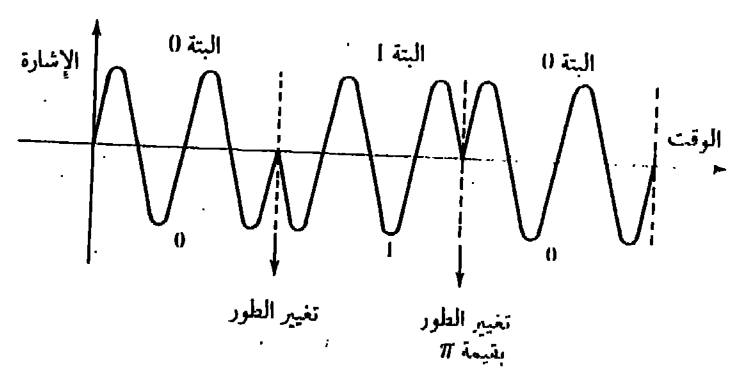
بنفس الطريقة التي نُضمّـن فيها سعة الموجة ، من الممكن أن نُضمِّـن بها الترددة . الأجهزة المستعملة لا تعمل في نفس الصيغة (شكل 6) .



شكل 6 ـ صيغة التضمين بالترددة

2.3.3 ـ التضمين الطوري

وفي النهاية ، من الممكن تضمين المُركّـب الثالث للموجة ، وطورها . الشكل 7 يبرهن الصيغة



شكل 7 ـ صيغة تضمين الطور

2.4 ـ الأخطاء في الإرسال

لا يوجد أي نظام إرسال بدون أخطاء : وبدون الكلام عن الأعطال البديهية مثل قطع الخط ، فعلى الخطوط يوجد إشارات طفيلية تؤثر على الإرسال وتغيّر الإرسال . ويجب إختزال أثرها إلى الحد الأدنى .

2.4.1 ـ المراقبة بالتكافؤ

الشكلان المشتركان في إكتشاف الأخطاء بالتدقيق بالتكافؤ هما التدقيق العامودي

(بالسمة) والتدقيق الأفقي (بالفدرة) .

التدقيق بالتكافؤ بواسطة السمة يتم بإضافة بتة إضافية إلى السمة . وإذا كان تعداد البتات '1' عند الوصول مفرداً ، يعني أن السمة قد جرى إستقبالها صحيحة . بعض الأجهزة يعمل بعكس هذه القاعدة .

السيئة الكبرى لهذه التقنية هي في حالة الأخطاء الشائعة على الخطوط (خط سيىء ، تبديل سيىء في منطقة مسار تلفوني مُكثّف) ، كما ولا يمكن إكتشاف الأخطاء المزدوجة .

لذا نضيف أيضاً سمة كاملة للتدقيق (BCC ، أو Block Check Character) في نهاية فدرة المعلومات . التدقيق سيتم خطاً بعد خط ، على جميع البتات في نفس المستوى .

2.4.2 ـ أكواد بنسبة ثابتة

في هذا النظام ، تُركّب مجموعة السمات بشكل تكون فيه نسبة عدد البتات «1» على عدد البتات «0» هي نفسها في كل سمة : كل إختلال في هذه النسبة يُمكن أن يتم كشفه .

مثلاً ، مجموعة السمات ذات النسبة 4 /3 من البتات «1» يُمكن أن نحصل عليها من خلال تكويد 7 أرقام ثنائية . عدد السمات المختلفة نحصل عليه بواسطة عدد المجموعات من n موضوع p إلى p .

$$C_n^p = \frac{n!}{p!(n-p)!}$$

الذي يعطي في هذه الحالة 35 ، إذا كان n هو العدد الكامل للبتات في السمة (هنا 7) وp هو العدد الكامل للبتات '1' (هنا 3 ، لأربعة بتات '0') .

2.4.3 ـ الأكواد الدورية

يتعلّق ذلك بأكواد اكتشاف الأخطاء الأكثر فعالية حالياً عند الاستعمال .

فلنفترض فدرة من المعطيات كعدد ثنائي صافي ونقسمه على عدد مُحدّد سابقاً . باقي القسمة يؤلف رقماً للتدقيق ، ويرسل مباشرة بعد فدرة المعطيات . عند الاستقبال ، نقسم فدرة المعطيات المتألفة من الفدرة الأصلية ومن رقم التدقيق على نفس العدد . وإذا لم يكن هناك خطأ ، فالباقي هو صفر .

3 ـ تقنيات إرسال المعطيات

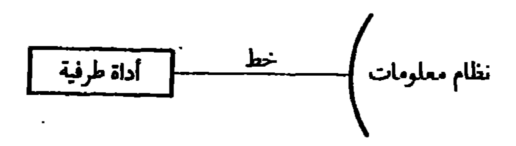
3.1 ـ الوصلات

3.1.1 ـ الوصلة نقطة إلى نقطة

يتعلّق ذلك بالوصلة الأسهل بين نظامين للمعلومات ، وبالتحديد بين الحاسب

والأداة الطرفية .

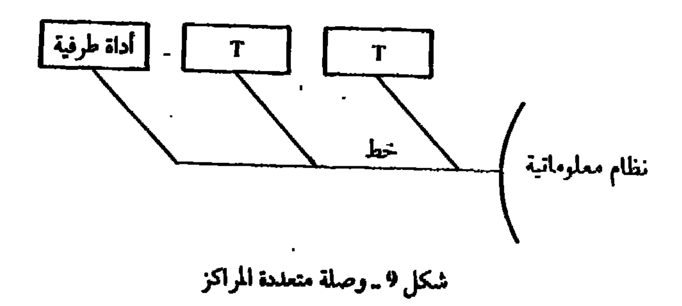
تبرّر هذه الوصلة إذا كان المسار مهماً : الخط غير المتحرّك يساوي ثمناً باهظاً.



شكل 8 ـ وصلة نقطة بعد نقطة

3.1.2 ـ وصلة متعددة المراكز

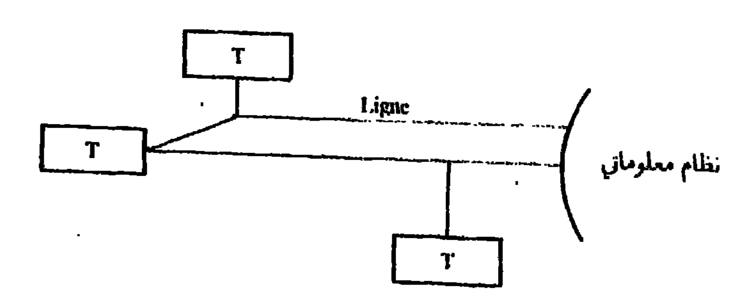
للاقتصاد في الخطوط ، يمكن تجميع عدة أدوات طرفية على نفس الخط (شكل 9) . نحصل على وصلة متعددة المراكز (النقاط) ، السيئة الوحيدة لهذه الوصلة ، هي في أن إنقطاع الخط الرئيسي يعزل عدة أدوات طرفية في نفس الوقت .



شكل 9 ـ وصلة متعددة المراكز

3.1.3 ـ وصلة حلقية

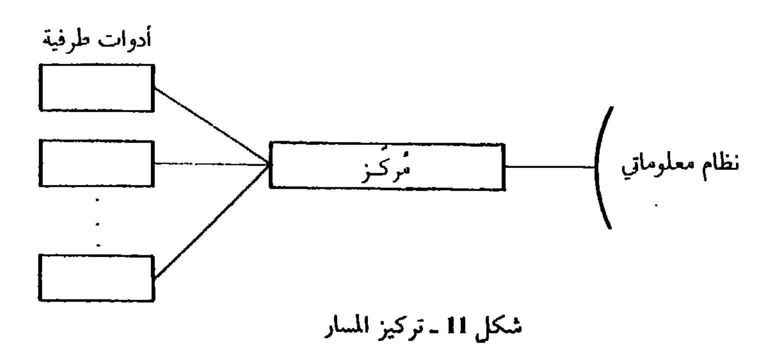
يتعلّق ذلك بحلّ وسط بين النظامين السابقين . تتصل كل محطة بالمحطة المركزية مرتين .



شكل 10 ـ وصلة حلقية

3.1.4 ـ مركز المسار

دائماً وبهدف تحسين تشغيل الخطوط ، التي ما برح ثمنها يزداد ، نقوم بتركيز المسار في عدة خطوط على خط واحد ، وبشكل عام بالنوعية الأفضل . نستعمل لهذا الغرض أدوات تدعى معابر أو مُركِّزات (شكل 11) .



شكل 11 ـ تركيز المسار

المعبر هو أداة لخلط الخطوط عند الانطلاق ولفصلها عند الوصول . المُركِّز يقوم بنفس الدور ولكن بطريقة سهلة : يتعلّق ذلك بحاسب ، مُتكيِّف مع بعض المهام الأخرى .

3.2 ـ التبديل

أساساً ، هناك نوعان جاهزان من التبديل : التبديل المكاني ، الكلاسيكي ، والتبديل المؤقت . الثاني ، وهو الأكثر استعمالاً ، هو بطبيعة معلوماتية .

3.2.1 ـ تبديل الدارات

التبديل المكاني هو تبديل الدارات ، والمصنوع منذ مدة طويلة للتلكس والتلفون . المراكز هي ميكانيكية وتستعمل مختلف الأنظمة . أحد الأنظمة الأفضل هو نظام Grossbar . سيئة هذا النظام هي في أنه يحتاج الى طلب دعوة ، قبل جميع المشاكل المحتملة والناتجة عن إشتغال الدارة . إضافة لذلك ، قد يحصل أن يحاول B طلب A في نفس اللحظة التي يحاول فيها هذا الأخير طلب B . وعلى العكس ، هناك فائدة للتبديل المكاني تكمن في الاتصال الفوري عند تنفيذ الطلب بشكل صحيح .

3.2.2 ـ تبديل الرسائل

تبديل الرسائل هي المرحلة الأولى من تقنية التبديل المؤقت . لا يوجد طلب ، على عكس النظام المكاني . الرسالة (سلسلة من السمات) تدخل إلى نظام التبديل ، مجهزة بعنوان وببعض عناصر المراقبة والتحكم . فلا نهتم بمعرفة ما إذا كانت المحطة المستقبلة

هي حرَّة أو مشغولة . تُخزَّن الرسالة في داخل النظام بشكل تكون فيه جاهزة للتسليم داخلياً .

للأنظمة بتبديل الرسائل بعض الحسنات على أنظمة التبديل المكانية :

- لا نخسر وقتاً عند إدخال الرسائل إلى النظام إذا كانت المحطة المُستقبلة مشغولة .
- الأدوات الطرفية ، غير القادرة على الإتصال فيما بينها بواسطة شبكة تبديلية ، لأسباب ناتجة عن عدم التوافق في السرعة أو في الكود ، يُمكن أن تقوم بالإتصال فيما بينها بواسطة نظام تبديل للرسائل : التحويلات تتم في مركز التبديل .
- الرسائل يُمكن أن تُسلَّم حسب أولويتها .
- لا يمكن إعادة إرسال الرسائل في حالة الخطأ .
- من الممكن الحصول على أفضلية عمل للدارات ، الخ .

تستعمل تقنية تبديل الرسائل حالياً ، وخاصة في مركز خدمات رسائل التلغراف الدولية . وفي بعض الحالات تستعمل في داخل الشركات على شبكاتها .

3.2.3 ـ تبديل الرزم

تقطَّع المعلومات الى رزم (Paquet) ، أو كميات منفصلة بنسق محكوم ، وبحجم قُصوي .

تحتوي الرزمة (شكل 12) على رأس ، مع معلومات التحكم والتسيير، وقسم للمعطيات بطول متحول ونهاية ، للتدقيقات .

بداية	عنوان	معلومات عن الطريق	معلومات إدارية	معطيات	تدقيق
8	16	32	32	0 à 2048	16 (bits)

شكل 12 ـ مثل عن الرزمة

هكذا إذاً يُمكن تقطيع رسالة إلى عدد من الرزم . ومع إن هذه الرزم تتمتع بنفس العنوان ، فيمكن معالجتها بشكل منفصل بواسطة النظام .

فوائد تبديل الرزم هي التالية :

- تبسيط أجهزة التدقيق والإتصال والمعالجة . من الممكن مثلاً إستبدال عدد كبير من

التوصيلات بسعة ضعيفة بواسطة إتصال بسعة كبيرة مربوطة بمركز معلوماتي . قد يكون هناك معايرة للنسق ، أو تبسيط لبرامج المعالجة .

- حماية ضد الأخطاء . التدقيق من نوع مُتعدِّد المخارج الدوري ، ويُمكن أن يُعاود في كل حلقة من المعالجة .
- تسيير أفضل . في كل حلقة معالجة ، يُمكن أن يكون النظام قادراً على حسابة الطريقة الأفضل للمتابعة لرزمة معينة .
- تحسين الخطوط الفيزيائية . هذه النقطة هي نتيجة للسابقة .
- تداخل وتحاور بين الأدوات الطرفية التي تعمل بسرعات مختلفة .

4 ـ عتاد المعلوماتية البعيدة

إحدى المسائل الأساسية ، في المعلوماتية البعيدة ، هي تلك الخاصة بمعايرة ونموذجة التوصيلات المختلفة المُركّبة للنظام .

بشكل خاص ، الملقى بين مُضمِّـن ـ كاشف والأداة الطرفية هو موضوع توصيات من جهة اللجنة CCITT ، وهذه التوصيات هي مقبولة بشكل واسـع من جهة مـراكز الخدمات في الاتصالات البعيدة إضافة إلى المُصمِّمين . وهي تُعرِّف الأنواع ، المهام ، عدد دارات التوصيل ومميزات الإشارات المتبادلة .

4.1.1 ـ عملية التبادل بين الموديم والأداة الطرفية

الأداة الطرفية «Terminal» هي عبارة عن حاسب لإدارة الإرسال : الاجراءات هي نفسها .

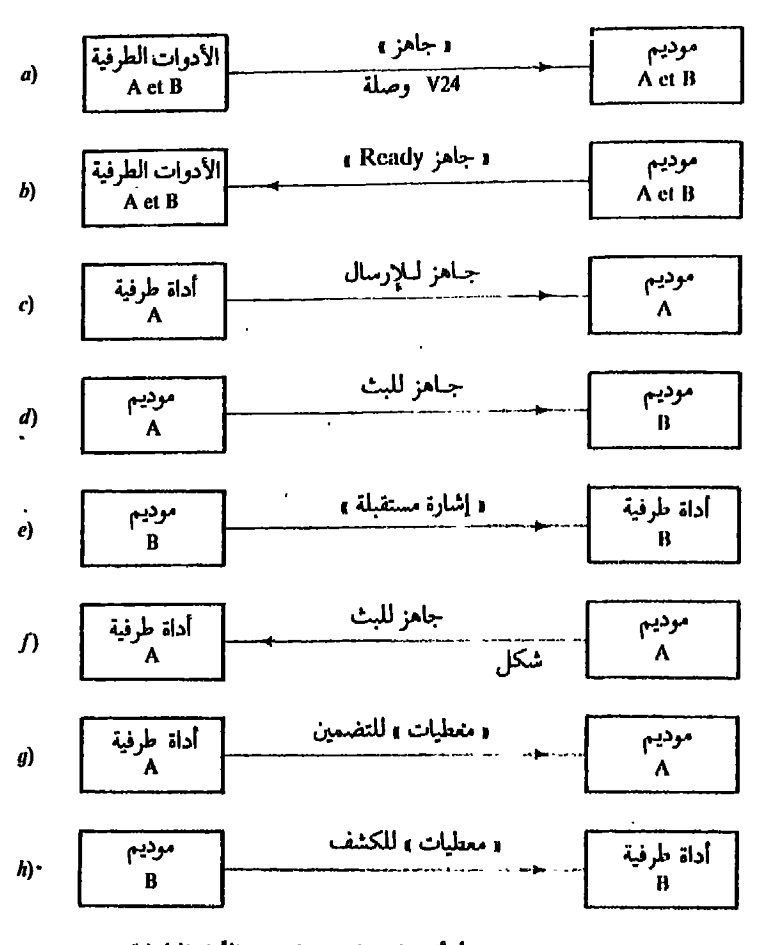
وبشكل تخطيطي تجري الأشياء على الشكل التالي :

أ ـ في كل طرف من الوصلة ، ترسل الأدوات الطرفية إلى الموديم إشارة « جهوز prêt » . وتنتظر منها إشارة شبيهة «prêt» في العودة .

ب ـ الموديم يُرسل بدوره إشارة « جاهز Ready » . هذه تتناسب أكثر مـع سؤال الأداة الطرفية : يتعلّـق ذلك بإرسال ، أو ببث الخ ؟

ج ـ في هذه الرحلة من العمليات ، لم يُحدَّد بعد إتجاه الإرسال الموديم A (مثلاً ، يجب أن يُوضع في البث والموديم B في حالة الإستقبال . المبادرة تعود إلى الأداة الطرفية A ، التي ستقوم بإرسال إشارة « جاهز للارسال Ready for transmission » إلى الموديم الخاص بها ، ماذا سيُفسِّـر الموديم (المضشف) بدوره :

د ـ المـوديم (مضشف) A يـرسـل إلى المـوديم B إشـارة « جـاهــز للبث Ready for emission » ، بشكل يُوضع فيه الموديم B في حالة الإستقبال ، ويُنبِّـه أداته الطرفية :



شكل 13 ـ مخطط مُبسّط للتبادل بين الموديم والأداة الطرفية

هـ ـ الموديم B يُنبّه فعلياً أداته الطرفية كي تصبح في موضع جاهز لاستقبال الإشارة .

و ـ الموديم A ، وبعد أن يكون قد إنتظر بعض الوقت (محدّد سابقاً) كي يصبح الطرف B في موضع جاهز للاستقبال ، يقوم بتنبيه الأداة الطرفية A بأنه قد أصبح جاهزاً للبث .

ز ـ الأداة الطرفية A ترسل المعطيات إلى الموديم ، الذي يقوم بتضمينها وإرسالها على الخط .

ح ـ الموديم B يستقبل الإشارات الآتية من A ، ويكشفها وينقلها إلى الأداة الطرفية B .

العملية تتتابع معكوسة للجواب ، وذلك بعكس دور A وB . نموذجة ومعايرة الملقى الفيزيائي بين الموديم والأداة الطرفية هو شرط أساسي لتوافق مختلف أنواع الأجهزة .

4.1.2 ـ الملقى V 24

التوصية CCITT V 24 تقوم على لائحة من دارات التبادل ، لكل دارة رقم يتألف من ثلاثة أرقام . السلسلة الأولى (السلسلة 100) هي مناسبة للدارات العامة . المجموعة الثانية (السلسلة 200) وهي مناسبة لدوال الطلب الأوتوماتيكية . ليست للأرقام أية علاقة مع الأجهزة الفيزيائية للتفريع على الموصلات ، تلك الأخيرة هي موضوع توصيات ISO (ISO / TC97 / SC6 n° 315)) .

الجدول التالي يُوجز الدارات الرئيسية للتوصية V 24 .

لائحة مختزلة بالدارات CCITT V 24

إسم الدارة	رقم الدارة
أرض	101
عودة الإشارة	102
معطيات مرسلة	103
معطيات مستقبلة	104
طلب إرسال	105
جاهز للبث	106
جهاز جاهز	107
موصل جهاز على الخط	108- 1
أداة طرفية جاهزة	108- 2
القنال «معطيات» إستقبال الإشارة	109
مكشاف لنوعية الإشارة	110
منقاة لسرعة الارسال	111
منقاة لسرعة الارسال	112
تزامن	113
تزامن	114
تزامن	115
إنتظار	116
دليل إنتظار	117
معطيات مرسلة	118
إستقبال المعطيات المرسلة	119
إرسال معكوس إشارة من الخط	120

قنال عودة جاهز	121
إستقبال إشارة على قنال عودة	122
مكشاف نوعية إشارة العودة	123
منقاة لمجموعة الترددات	124
دليل الطلب	125
منقاة لترددة الإرسال	126
منقاة لترددة الاستقبال	127
تزامن	128
معطيات إستقبال	129
إرسال عودة	130
استقبال لسمة تزامن	131
عودة في صيغة لا معطيات	132
جاهز للاستقبال	133
معطيات مستقبلة	134

4.2 ـ الموديم (مُضمِّن ـ كاشف ـ مضشف)

الموديم ، وكما رأيناه ، هو عبارة عن جهاز قادر على تضمين إشارة مربعة خارجة من أداة إرسال بالتوالي ، بشكل تصبح معه هذه الإشارة قابلة للنقل على خطوط بمسافة بعيدة .

وهو يقوم أيضاً بالعملية المعاكسة .

ويوجد عملياً عدد كبير من أنواع الموديم على قدر ما يكون هناك أنواع من التضمين : بالسعة ، بالترددة أو في السطور .

بنفس الطريقة ، قد يكون هناك موديم بسرعة عالية أو منخفضة تتناسب مع جميع إمكانيات وسائط الاتصال ، من 300 بود حتى 48 كيلو بود وأكثر .

الموديم بسعة منخفضة تتوافق بشكل عام مع التوصية V 21 للجنة CCITT وتسمح بالعمليات المتزامنة بسبب إستعمال نطاقين للترددات :

نطاق 1 :

1180 هرتز Hz ('0' ثنائي)

980 هرتز Hz ('1' ثنائي) قنال منخفض

نطاق 2 :

1850 Hz ('0' ثنائي)

1650 Hz ('1' ثنائي) قنال أعلى

توصيل الموديم على الخط يمكن أن يكون مباشراً، أو غير مباشر . في الحالة الأخيرة ، نستعمل وصلات صوتية ، تسمح بإرسال ترددات على الخط التلفوني دون أن يكون هناك حاجة للمس أي شيء من الأجهزة المركبة . يكفي أن نضع التلفون على ناقل خاص مركب على الموديم كي تصبح الوصلة جيدة . الأهمية الكبرى لهذا الجهاز هو في أنه يسمح بالتوصيل على شبكة في أي مكان . وهذا هو الذي سمح بتطوير الأداة الطرفية المنقولة .

4.3 ـ المعابر (Multiplexor)

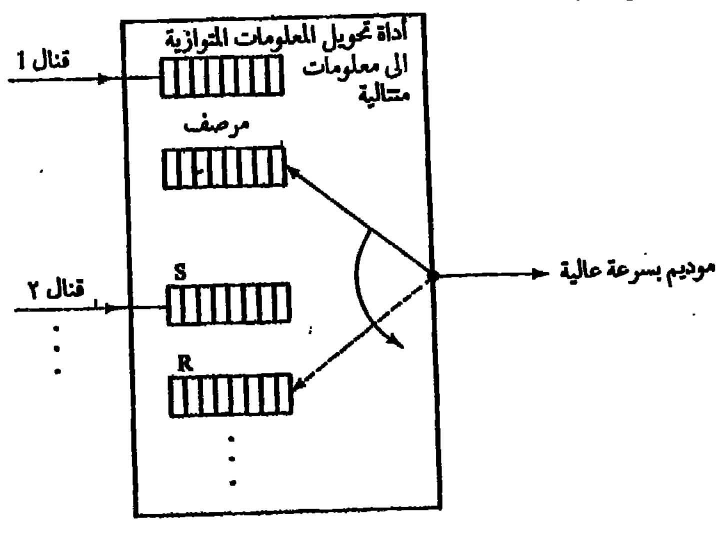
الهدف الرئيسي للمعبر هو في إختزال ثمن الخطوط ، وذلك بتحسين حملها . نستعمل خطاً تلفونياً من 2400 أو 4800 بود ، أي ثميناً، كي نقوم بإجراء عدة وصلات بسرعة منخفضة (مثلاً 200 بود) ، أكثر من تحريك عدة خطوط من 200 بود (إفرادياً أثمن) ، باستعمال ثانوي : المجموعة من هذه الأخيرة هي أغلى من خط تلفوني واحد .

هناك نوعان من العبور : في الزمن وفي الترددة .

4.3.1 ـ العبور الزمني

الصيغة هي التالية : نأخذ المعطيات من عدد من المصادر ونُخصِّص لكل مصدر دورة زمنية معينة . كل دورة من هذه الدورات الزمنية تجمع مع الأخرى لتشكّل سلسلة من المعطيات التي سيتم إرسالها بسرعة عالية .

قد يحدث عبور للبتة أو السمة . في الحالة الأخيرة مثلاً ، تُستقبل المعطيات على التوالي في أداة تحويل المعلومات المتوازية إلى معلومات متتالية ، والآتية من جهاز بسرعة منخفضة (شكل 14) .



شكل 14 ـ صيغة العبور الزمني

4.3.2 ـ العبور بالترددة

في صيغة العبور بالترددة ، أو بقسمة الترددات ، نُقسِّم النطاق العابر الجاهز على الخط إلى عدد من النطاقات المستقلة ، مثلاً ، النطاق العابر التلفوني (300 هرتز إلى 3400 هرتز) يمكن أن يُقسَّم الى 12 قنالاً من 240 هرتز كل منها . لهذا السبب بإمكان 4 خيوط أن تقوم بإجراء 12 وصلة من نوع full-duplex بسرعة منخفضة .

4.4 ـ المركزات Concentrateur

المركزات هي عبارة عن حاسبات صغيرة مُخصَّصة لإدارة الخطوط . بينما تُقدِّم المعابر الحالية صلابة في الاستعمال ، فالمركزات ، وهي مبرمجة ، تبدو وكأنها أكثر ليونة وبساطة .

صيغة التشغيل تبقى نفسها : نقوم بالعبور المؤقت . ونُسجِّل مؤقتاً سمات أو رسائل في مراصف بسرعة منخفضة . تنقل هذه السمات أو الرسائل على التوازي ، إلى مراصف أخرى أو إلى حيزات من الذاكرة من حيث هي مستخرجة وبسرعة كبيرة كي ترسل على مخرج واحد .

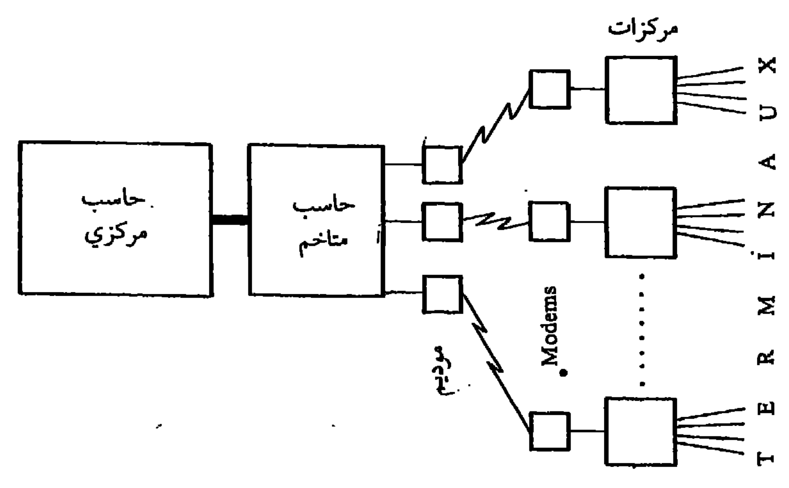
فوائد المركزات هي التالية :

- تتكيَّف بسهولة مع سرعات وإجراءات جميع أنواع الأدوات الطرفية ، لأنها تبرمج لهذا الغرض .
- تسمح ببسط مؤشرات المسار :
- يمكنها إكتشاف وتصحيح الاخطاء في الإرسال أوتوماتيكياً .
- يمكنها إجراء معالجات ، معقدة ، على المعطيات المطلوب إرسالها : ترجمة (مثلاً في ASCII أو EBCDIC تغيير النسق ، توليد الإطارات ، إختبار الصلاحية الخ .
- يمكنها أن تدير إشتغال الأدوات الطرفية ، والمحافظة على الأشكال في حالة العطل في الحاسب المركزي .
- يمكنها أن تجمع إحصائيات من جميع الأنواع في مسار المعطيات ، وتجميع المعطيات «off-line» لإعادة الإرسال الداخلي ، الخ .

4.5 ـ الحاسبات المتاخمة

الحاسبات المتاخمة (Front-end processors) هي عبارة عن حاسبات موضوعة قبل حاسبات المعالجة الكبيرة ، كملقى بين هذه الأخيرة وبين شبكة معلوماتية بعيدة . وتلعب دور المركزات ، وتؤمن بعض الأعمال الإضافية (شكل 15) .

توضع الحاسبات المتاخمة لمساعدة الحاسبات في معالجة الأعمال الخاصة بالمعلوماتية البعيدة .



شكل 15 ـ حاسب متاخم (شبكة معلوماتية)

وتدير الحاسبات المتاخمة الخطوط والأدوات الطرفية ، وذلك بإدارة خطوط التلغراف والتلفون مع أو بدون موديم ، والاشراف على الإجراءات : إدخال أو إلغاء سمات ، إكتشاف ومعالجة الأخطاء ، تعليق عمليات المراجعة . كما وتؤمن ، محتملاً ، التكيّف الأوتوماتيكي مع سرعة الأدوات الطرفية اللاتزامنية ، وذلك باعتماد السرعة الكبيرة وبالمقارنة مع سمة نموذجية كمعيار . كما وتؤمن أيضاً مهام العنونة .

ـ تدير الاتصال مع الحاسب المركزي . الوصلات يمكن أن تكون بعدة أنواع : قنال إلى قنال ، بلوغ للذاكرة المركزية ، الخ .

ـ يمكن أن نقوم باستغلال كامل للشبكة : حوار خدمة مع الأدوات الطرفية ، تخزين الرسائل ، تحميل بعيد للمركزات ، إدارة عمليات التوقف ، الخ . كما ويمكن في بعض الحالات استبدال الحاسب المركزي في حالة حدوث عطل أو توقف بغرض صيانة هذا الأخير .

ـ وبإمكانها أخيراً معالجة الرسائل ، ولكن من الأفضل ترك هذه المهمة للمركزات . وبالتحديد في كل ما يتعلق بإطار هذه الرسائل وتركيزها (في الاتجاه مركزات ـ أدوات طرفية) أو في رص السمات بالتوكيد (في الحالة المعاكسة) . هكذا فمن المفيد إختزال كثافة المسار على الخط من مسافة بعيدة ، على كل حال لن نهتم بهذه المشكلة في مستوى الوصلة المتاخمة للحاسب المركزي .

وفي النهاية بإمكانها تشكيل خطوط انتظار على الأسطوانات .

5 ـ الأدوات الطرفية (TERMINAL)

نفهم بالأداة الطرفية ،جهازاً قليل أو كثير التعقيد للادخال . الاخراج بسرعة منخفضة أو بسرعة وسطية ، ومُعتمد كعضو أساسي في الحاسبات : ذاكرة ودارات معالجة عامة .

وما يزال سوق الأدوات الطرفية يتطور بدون إنقطاع ، ويتصرفنا الآن أدوات طرفية تدعى « ذكية » ، هي عبارة عن حاسبات صغيرة ، قادرة على العمل محلياً ومركزياً كحاسب .

ولقد توسّع سلّم الأدوات الطرفية كثيراً ، ولكن من الممكن تجميعها في بعض الأنواع الكلاسيكية :

5.1 ـ الأدوات الطرفية بالطابعة

هذه الأدوات الطرفية ، وكأغلب الأدوات الطرفية الخفيفة ، هي مزودة بلوحة ملامس . إضافة لذلك فهذه الأدوات الطرفية هي مزودة بطابعة ، قضيبية (كمكنة الكتابة العادية ، بكرة (أدوات طرفية IBM بشكل خاص) ، بالأبر . . . الخ .

سرعة هذه الأدوات الطرفية هي محدودة بالميزات الميكانيكية للطابعات : ونجد السلّم التالي منها :

- 10 سمات في الثانية (أغلب هذه الأدوات) .
- 15 سمة في الثانية (بعض أدوات IBM الطرفية) .
- 30 سمة في الثانية وحتى 45 سمة في الثانية (طابعات هذه الأدوات الطرفية هي قادرة على الإرسال ، والاستقبال بسبب ملامسها .

بعض هذه الأدوات هو مُجهّز بأدوات إرسال أوتوماتيكي ، مغناطيسي أو ناقل ورقي (شريط مثقوب) . وبإمكانها أن تقوم بالإرسال بواسطة وسائط أخرى غير لوحة الملامس . تمتاز بعض الأدوات الطرفية بواسطة ذاكرة مكدس (Buffer) ، تسمح بتسريع التشغيل . وبشكل خاص ، تستعمل مدة عودة المجر لتخزين السمات الآتية من الموديم . هكذا ، فسرعة الأداة الطرفية يُمكن أن تكون أعلى بشكل قليل من تلك المناسبة لسرعة الخط .

5.2 ـ الأدوات الطرفية بطباعة حرارية أو غير ذلك

هذه الأدوات الطرفية لا تختلف عن السابقة إلا بنوع الطباعة المستعملة . يتعلّق ذلك إما بتكويد كيمائي لغلاف الورق وإما بإسقاط حبر دقيق عليه .

في هذه الأداة الطرفية نلاحظ إن الأواليات الميكانيكية هي قليلة ، ومن الممكن أن تصل إلى سرعات عالية ، إلى 120 سمة في الثانية وأكثر فائدة .. هذه الطابعات هي في عدم

إصدار أصوات عنها ، ولكن السيئة الكبرى فيها هي في إستعمال ورق خاص (بالأنظمة الحرارية) . إضافة لذلك فإن إمكانية رؤية وقراءة الأحرف هي أقل جودة منها في الأدوات السابقة .

5.3 ـ الأدوات الطرفية بالشاشة

هي الأدوات الطرفية ، . حيث عضو الإخراج هو عبارة عن أنبوب كاتودي معنون بالنقاط (أدوات طرفية للرسم) ، أو إبجعددي . ولكن لوحة الملامس هي نفسها .

في الحالة الأولى ، تؤدي المعلومات المُستقْبَلة إلى التركيز والتحريك من نقطة إلى أخرى : من الممكن أيضاً الرسم ، أو عند الحاجة الكتابة ، ورسم أي شكل نرغب به . الخ .

في الحالة الثانية ، تولّد السمات المُستقْبَلة على الشاشة ، بدلاً من أن تُضرب على الورق . ولا يوجد أي فرق مُحدَّد بالنسبة للأدوات الطرفية بالطابعات .

وفي أغلب الأحيان ، يمكن لبعض الأدوات الطرفية ، أن تُزوَّد ، وبسبب ذاكرة الداريء المستعملة ، إلى سهولة في التنقيح : محو السمات أو الخطوط ، جدولة ، إدخال خطوط إضافية أو سمات ، محو كامل لصفحة واحدة . . .

الأدوات الطرفية بشاشة هي صامتة بمجملها ، وهي ذات فائدة كبيرة في الاستعمال والاستيفاء اليومي والعمل في الوقت الفعلي على مجمع للمعطيات . إضافة لذلك ، فسرعة العرض تجعلها كثيرة الفعالية . لهذا السبب نراها واسعة الانتشار وخاصة في التطبيقات العملية الإدارية والمؤتمتة في الوقت الفعلي ، وهي مفضّلة على الأدوات الطرفية بالطابعات .

5.4 ـ أدوات طرفية أخرى

لقد ظهرت مجموعة غير محدَّدة من الأدوات الطرفية ومن جميع الأنواع خاصة بالمعلوماتية البعيدة : كل شيء ممكن في هذا الحقل .

فلنذكر بعضها : الأدوات الطرفية للتدقيق بالعلامات ، صناديق التسجيل ـ الطرفية في المخازن ، الأدوات الطرفية اللاقطة للمعلومات الفيزيائية (حرارة ، ضغط ، سرعة ، الخ) . الخ . .

6 ـ إجراءات إدارة الخطوط

إجراء إدارة الخط هو بشكل أساسي خوارزم للحوار بين نظامين للمعلومات من

خلال الخط . لا يوجد معلوماتية بعيدة إذا لم يكن بمقدرة النظامين « التفاهم » أي إستعمال بروتوكول لتبادل الرسائل(*) .

وهذا يؤدي إلى وجود ، في كل نظام ، إلى مناهج خاصة ، تخضع لنفس القواعد تحديداً أي لنفس البروتوكولات .

ننصح كثيراً بأن تكون هذه البروتوكولات نموذجية وموضع معايرة ، لأسباب تجارية بديهية : يجب على الأداة الطرفية أن تقدر على الاتصال بأي شبكة . وبشكل أكثر عمومية ، يجب على كل حاسب أن يقدر على تبادل المعطيات مع أي حاسب آخر من الشبكة . هناك محاولات للمعايرة والنموذجة : مثلاً تلك التي تتعلق بالاجراء H.D.L.C (**) .

6.1 « نصف إزدواجي » و« إزدواجية كاملة » (Half duplex, full-duplex)

بالنسبة لخط فيزيائي معين ، فمن الممكن إجراء عملية الإرسال في الإتجاهين ، أي الإرسال والاستقبال في نفس الوقت ، أو إستعمال إجراءات تدعى full-duplex أو أيضاً ، على العكس ، عملية الإرسال تتم في إتجاه واحد في لحظة معينة ، وبعد ذلك في الإتجاه التالي ، ومن ثم نتبع إجراءات تدعى half-duplex أي نصف إزدواجية (bilatéral à l'alternat) .

على نفس الخط الفيزيائي ، الإرسال المتزامن في الإتجاهين يُمكن أن نحصل عليه باستعمال نطاقين منفصلين من الترددات . يُستعمل النطاق الأول في الاتجاه الأول ، والثاني في الاتجاه الآخر . ليس من الضروري أن يكون عرض النطاق هو نفسه في الحالتين : النطاق الأعرض سيتم إستعماله لارسال المعطيات ، والأصغر لتبادل إشارات التحكم مثلاً .

6.2 ـ التزامن

إرسال المعطيات يُمكن أن يكون تزامنياً أو لا تزامنياً . يُمكن أن يحصل التزامن في ثلاث مستويات .

- في مستوى البتة .
- في مستوى السمة .
- في مستوى الرسالة .

6.2.1 ـ الإرسال المتزامن

الإرسال المتزامن ، ويدعى أيضاً إنطلاق ـ وقف (Start-Stop) ، ويؤمن التزامن في مستوى السمات . تلك الأخيرة يُمكن أن ترسل الواحدة بعد الأخرى على الخط وفي

(*) وهذا ينتج عن المعلوماتية بشكل عام .

(**) High level Data Link Control : H.D.L.C

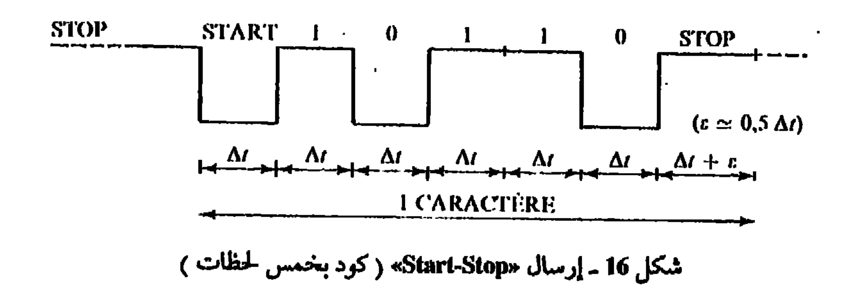
لحظات مختلفة : المدة التي تمر بين إرسال سمتين مختلفتين بدون أية أهمية .

نستعمل هذه الطريقة كثيراً في الطابعات البعيدة ، وفي الأدوات الطرفية البطيئة المجهّزة بلوحة ملامس .

يدخل العنصر «Start» (شكل 16) في بداية كل سمة .
ويتألف من '0' ثنائي ، أو « فراغ » . العنصر «Stop» يدخل أيضاً في نهاية السمة .
ويقوم على «1' ثنائي ، أو « علامة » . يحتفظ بالعنصر «Stop» بين كل سمة : وليس له أية مدة محددة .

عندما يكتشف النظام المعلوماتي (مثلاً الأداة الطرفية) أي تغيير في الأقطاب ، تبدأ ساعة مركزية العمل ، فتراقب لحظات معايرة (échantillonnage) الخط .

وعندما يتم إستقبال العنصر «Stop» ، تقف الساعة ، وتنتظر إشارة «Start» جديدة .



شكل 16 ـ إرسال «Start-Stop» (كود بخمس لحظات)

الإرسال اللاتزامني يستعمل إما كود بودوا (Baudot) بخمس لحظات ، وإما ، بشكل عام الألفباء ASCII . هذه الأخيرة (ASCII) تتألف من 8 لحظات ، إضافة إلى العنصر «Start» ، والعنصر «Stop» الذي يتألف بدوره من عنصرين نموذجين من التشكيلة '1' الثنائية المتتالية .

6.2.2 ـ الإرسال اللاتزامني

نعتبر التقنيه «Start-Stop» صالحة بالنسبة للأدوات الطرفية البطيئة ، ولكنها تصبح غير فعالة إذا أردنا أن نقوم بالإرسال في سرعات مرتفعة

في الإرسال التزامني ، تتبع بتات إحدى السمات مباشرة بيتات الأخرى . لا يوجد لا 'start' ، ولا 'stop' بين السمتين .

على العكس ، يقسّم دفق السمات الى فدرات ، جميع البتات من الفدرة ترسل في

فسحات بأوقات متساوية . ويجب أن يكون النظامين ، المُرسل والمُستقبل متزامنين بالكامل .

خلال طول مدة إرسال الفدرة . يجري التحكُّم بهذا التزامن بواسطة السمات الأولى من الفدرة : وبشكل عام بالسمات «SYN» ASC (التشكيلة الثنائية '00010110') .

6.3 ـ المعالجة H.D.L.C

سنأخذ الاجراء H.D.L.C ((High level Data link Control) أو إجراء التحكُّم بالوصلة الرقمية بالمستوى العالي ، كمثل على إجراء التبادل (*) . عناصر المعايرة هي بشكل أساسي تركيبة السلّم . (ISO DIS 4335). معايرة الاجراءأصبحتضرورية نظراً للانتشار السريع للمعلوماتية البعيدة وظهور الشبكات المعلوماتية لإرسال المعطيات والخاصة بالمواطنين .

6.3.1 ـ تركيبة الشبكة

في الأجراء H.D.L.C تتم جميع عمليات الإرسال بداخل شبكات بالشكل الموجود على الشكل 17 .

FANION 01111110	عنوان (8 bits)	COMMANDE (8 bits)	معلومات	F.C.S. (16 bits)	FANION 01111110

شكل 17 ـ السلّم H.D.L.C

Fanion : تبدأ كل شبكة وتنتهي بالمتتالية '01111110' . هذه المتتالية يبحث عنها دائماً بواسطة المحطات المتعلقة بالشبكة . وتستعمل هذه المتتالية لتزامن السلّم .

Adresse : العنوان الذي يُعرِّف عن المحطة التي يُوجّه إليها الأمر أو المحطة التي ترسل الجواب .

Commande : يحتوي حيز الأمر على معلومات تدل على الأمر أو على الجواب . ويحتوي أيضاً على رقم السلّم (trame) المستعمل لمراقبة الإرسال .

Fram Control sequence F.C.S. ، أو متتالية التدقيق بالسلّم : سلسلة من 16 عنصراً ثنائياً تستخدم للتحقق من صلاحية السلم بواسطة طريقة متعدّدة المخارج .

(*) المعلومات المتابعة هي خارجة عن وثائق ISO . ولا يجب أن تعتبر نهائية ، الأجراء H.D.L.C لم يكن موضوع معايرة حتى تاريخ ظهور هذا الكتاب .

الميزة المهمة في تركيبة السلّم هي الشفافية لجهة المعطيات المرسلة . لهذا ، من الضروري في كل قسم من السلّم الموجود بين 2 «fonion» إلغاء التشكيلات الثنائية التي تمثّل أل fanion . الوسيلة المختارة تقوم على إدخال بتة '0' عند الإرسال بعد كل متتالية من 5 بتات '1' . وعلى العكس ، عند الاستقبال نلغي كل بتة '0' حسب متتالية من 5 بتات '1' .

6.3.2 ـ الاجراءات من نوع H.D.L.C

المعيار .H.D.L.C لا يعرِّف أبداً نوعاً واحداً من الاجراءات ، ولكن بواسطة مجموعة من الأوامر والأجوبة ، بإمكاننا تعريف عدد من الإجراءات المُتكيفة مع تشكيلات مختلفة .

6.3.2.1 ـ نوع المحطات

يُعرَّف الأجراء في معنى الأوامر المرسلة بواسطة محطة أولية وأفعال وأجوبة المحطة أو المحطات التي تدعى ثانوية . لتعميم ذلك سنقوم بإدخال نوع من المحطات مختلط يمكنه إرسال أوامر وأجوبة في نفس الوقت .

6.3.2.2 ـ طرق العملية

عندما تكون الوصلة فعالة ، نستطيع تمييز طريقتين أساسيتين في العمل . في الصيغة العادية ، لا يمكن للمحطة أن ترسل سلالم تحتوي على معلومات إلا بإذن من المحطة الأولية . في الصيغة التزامنية . يُمكن للمحطة أن تبدأ بإرسال المعلومات دون السماح الواضح للمحطة الأولية .

6.3.2.3 ـ شكل حقل الأمر

تعريف نوع الأمر أو الجواب يتم في حقل الأمر حسب الشكل المعطى في الشكل 18 .

نوع الرسالة	حيز الأمر 0	1	2	3	4	5	6	7	ملاحظة
I تبادل معلومات	0	N(S)			P/F	N(R)			I. N(S). N(R). P/F
S أمر / جواب مراقبة	1	0	S	S	P/F	N(R)			S. N(R). P/F
U أمر / جواب غير مرقم	1	1	M	M	P/F	M	M	M	M. P/F

شكل 18 ـ شكل حيز (حقل) الأمر

N (S) رقم المتتالية عند البث (الإرسال) (من 0 إلى 7)

N (R) رقم المتتالية عند الاستقبال (من 0 إلى 7) .

S تعريف الأمر أو الجواب التدقيقي .
M تعريف الأمر أو لجواب الغير مُرقّـم .
P / F بتة «Palling» (دعوة للبث) للأمر .
بتة نهاية (نهاية الإرسال) للجواب .

6.3.2.4 ـ الترقيم المزدوج المستقل

تستعمل الاجراءات من نوع H.D.L.C لمراقبة الإرسال والعودة الإستثنائيـة عند حدوث خطأ ، طريقة تدغى مزدوجة الترقيم ومستقلة . تحافظ كل محطة على متحـولة إرسال S ومتحولة إستقبال R ، كل منها مُعدّ ومصفّـر . للمحطة الأولية ، هناك عدد من الأزواج R, S يعادل عدد المحطات الثانوية . صيغة عمل النظام هي التالية :

ـ إرسال سلّـم من نوع I .
إجعل (modulo 8) N(S) = S و S = S + 1
إحتفظ بالسلّـم لاعادة الإرسال المحتملة .

ـ إرسال سلّـم من نوع I أو S .
إجعل N (R) = R

ـ إستقبال سلّـم من نوع I صالح (بدون خطأ F.C.S) .
إذا N (S) = R إجعل (Modul 8) R = R + 1 وعالج السلّـم وإلا إطلاق معالجة العودة للعمل .

ـ إستقبال سلّـم من نوع I أو S صالح .
حرّر السلم المرسل برقم أقل من N (R) .

وبإيجاز ، تُرقم كل محطة سلالمها بواسطة متحولة S ، بينما المتحولة R تدل على الرقم N (S) للسلّـم التالي (ما يُنتظر إستقباله في المحطة التالية) . هذه المتحولة R تزيد في كل إستقبال من السلّـم الصحيح وإرسالها بواسطة N (R) لسلّـم من نوع I أو S يُبرّر السلالم I برقم أقل من N (R) المرسلة بواسطة المحطة الأخرى . إجراء المعاودة يقوم أساساً عـلى طلب إعادة إرسال السلم الناقص ذو الرقم R بواسطة سلّـم خاص S' ، وذلك إما بشكل معزول وإما متبوعاً بجميع السلالم المرسلة بعدها .

6.3.2.5 مثال على الحوار

الشكل 19 يعطي مثلاً على الحوار في الإجراء H.L.D.C.
الوصلة هي من نوع «full-duplex» وطريقة العملية هي الطريقة العادية التزامنية .
سنلاحظ استعمال البتة P/ F بواسطة كل من المحطات . الملاحظات هي تلك الخاصة بالشكل 19 . ومجموعة الأوامر والأجوبة المستعملة هي التالية :

إرسال للمعلومات بواسطة المحطة الأولية فقط

أولية: SNRM, P — I, O, O — I, 1, O — I, 2, O, P — I, 3, O

ثانوية: UA, F — RR, 3, F

إرسال المعلومات بواسطة المحطة الثانوية فقط

أولية: SNRM, P — RR, O, P — RR, 3

ثانوية: UA, F — I, O, O — I, 1, O — I, 2, O — I, 3, O — I, 4, O, F

إرسال المعلومات بواسطة المحطتين (مع خطأ في الإرسال)

أولي: SNRM, P — I, O, O, P — I, 1, O — I, 2, O — I, 3, 1 — I, 4, 2 — I, 2, 2

Erreur de transmission — trames retransmise

ثانوي: UA, F — I, O, 1 — I, 1, 2 — REJ, 2 — I, 2, 2

Demande de retransmission

شكل 19 ـ مثال على الحوار ، الصيغة العادية ، **full-duplex** (في الأجراء H.D.L.C)

نوع S	RR, N(R)	Receive Ready . يدل على أن المحطة هي جاهزة لاستقبال السلالم من نوع I . يسمح بتبرير السلالم بسبب N (R)
	REJ, N(R)	Reject . طلب إعادة الإرسال لجميع السلالم من نوع I المرسلة بواسطة المحطة الأخرى وبرقم > = N (R)
نوع U	SNRM	Set Normal Response . ضع الثانوي Mode في موضع الصيغة العادية .
	UA	Unnumbered Acknowledge طلب إستقبال غير مُرقّم لأمر بواسطة الثانوي .

7 ـ شبكات المعلوماتية البعيدة

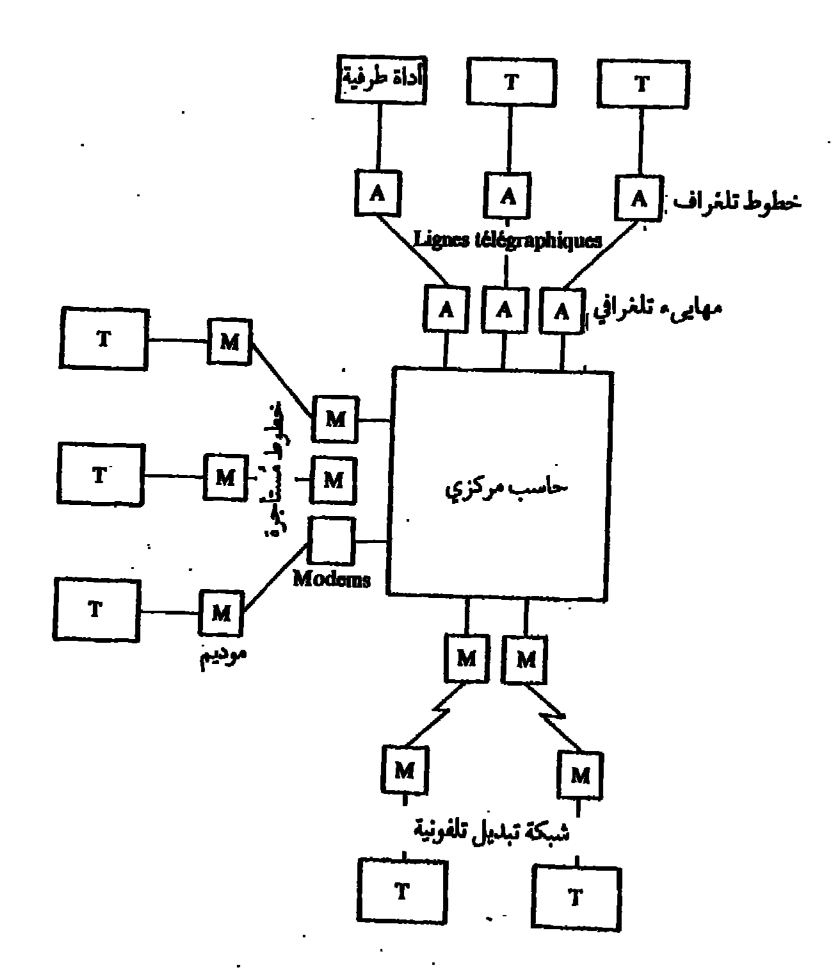
يوجد أنواع عديدة من شبكات الحاسبات ، أو شبكات المعلومات البعيدة . بعض هذه الشبكات هو بسيط ، متجانس ، أو بحاسب وحيد. وعلى العكس ، هناك شبكات غير متجانسة وتربط عدة أنواع فيما بينها . والآن بدأت التقنيات المناسبة للشبكات توضع في

العمل ، ونصرّ الآن على تركيبات ، إن في العقل التجاري أو في مستوى الإدارات الكبيرة .

7.1 ـ الشبكات البسيطة على شكل نجمة

تتألف هذه الحاسبات بشكل أساسي من حاسب ، حيث تربط بها مباشرة بعض الأدوات الطرفية (شكل 20) .

يُمكن للأدوات الطرفية أن ترتبط بالحاسب المركزي بطرق مختلفة : الوصلة الأكثر إستعمالاً تتم بواسطة موديم . نستخدم غالباً شبكات التلفون العامة لأجراء وصلة فيزيائية بين الموديم : السرعة الإسمية لهذه الوصلة هي عادة أقل مما هـو ضروري (2400 بـود لاستعمال مقوي يعادل 300 بود ، أي إلى 30 سنمة في الثانية تقريباً) ولكن الفائدة تكمن في



شكل 20 ـ شبكة بسيطة نجمة

عدم تقييد الخط الفيزيائي إلا في الوقت الذي يجري فيه الإتصال بالحاسب . وهذه هي الحالة المثلى للتطبيقات في Time-Sharing ، مثلاً .

من الممكن أيضاً أن نجد وصلات ثابتة ، بواسطة خط بسيط (أي بشريطين) مُستأجر . وهذا لا يُبرَّر إلا إذا كانت مدة التوصيل الفعلية كافية ، أو عندما نكون مجبرين للعودة إلى هذا الحلّ ، إذا كانت نوعية الشبكة المبدَّلة غير كافية . ومن الممكن في النهاية إستعمال وصلات تلغرافية من 200 بود ، وفي خطوط مستأجرة ، بثمن أقل من الخطوط التلفونية .

7.2 ـ شبكة متعددة نجمة

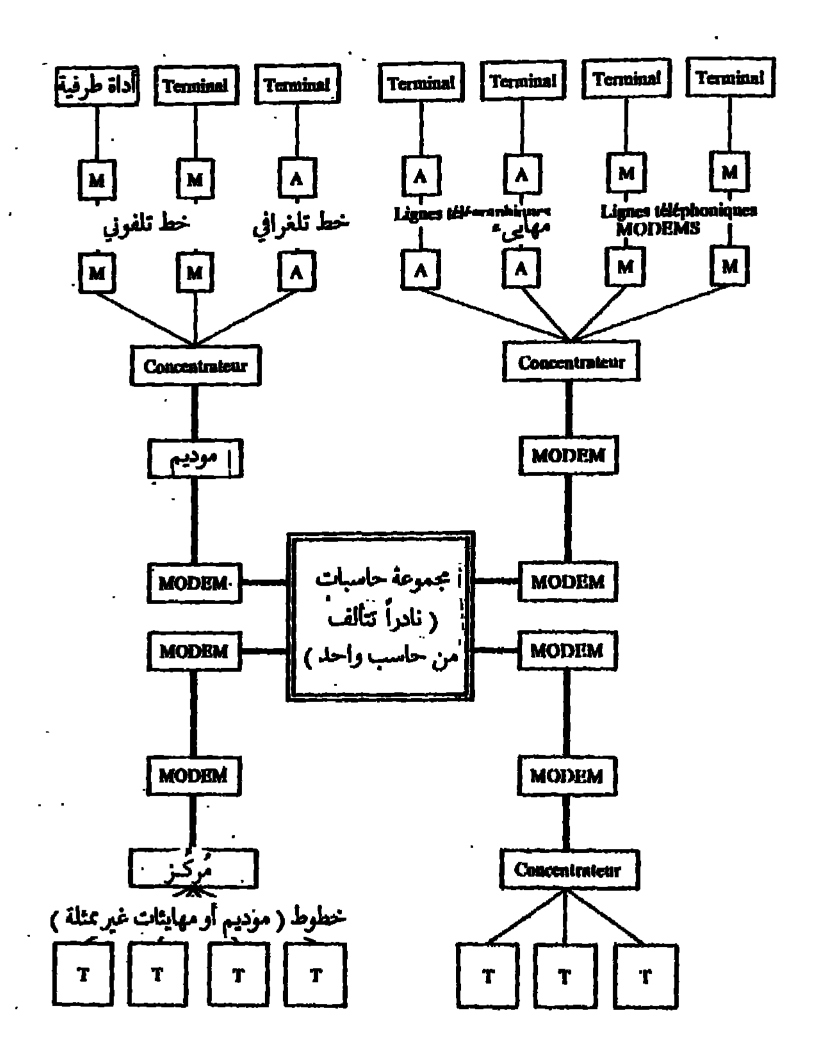
عندما يستدعي التوسيع الجغرافي لشبكة بسيطة « نجمة » أن الخروج عن إطار المنطقة ، فمن الأفضل إستعمال التركيبة التي تدعى نجمة متعددة (شكل 21) .

صيغة التشغيل هي نفسها كما في الحالة نجمة بسيطة ، ولكن نجد بعض المُركِّزات في ملقى بين الحاسب المركزي والأدوات الطرفية .

يُربط الحاسب المركزي بالمُركِّزات بواسطة خطوط ثابتة ، مُستأجرة بسرعة كبيرة (من 1200 إلى 4800 بود ، وقد تصل في بعض الحالات حتى 84 كيلو بود) ، تُربط الأدوات الطرفية جميعها بالمُركِّزات بواسطة وسائط شبيهة بالتي رأيناها في الفقرة السابقة بالنسبة للشبكة البسيطة نجمة .

شبكات الشركات ، في فرنسا ، هي غالباً من هذا النوع عندما يكون هناك توسيع وطني . لا شيء يمنع من أن يصبح التوسيع بالمعيار الدولي . من الممكن جيداً أن نجد حاسباً للمعالجة في الولايات المتحدة ومُركِّز في أوروبا ، الوصلة بين الحاسب والمركز تتم بواسطة وصلة عبر المحيطات ، أو بواسطة الأقمار الإصطناعية . هكذا هي مثلاً الشبكة التجارية MARK III . عندما تكون الشبكات بهذا التوسع ، فمن الجاري أن نجد عدة حاسبات متشابهة في مركز النجمة الرئيسي ، لأسباب ناتجة عن الحماية . ويمكن عند ذلك توزيع حمل الحاسب الدائم على الحاسبات الأخرى .

البلوغ بواسطة الأداة الطرفية للحاسب المركزي يتم بواسطة تفريع على المركز الأقرب . في حالة الشبكات الكبرى الدولية هناك واحد أو عدة مُركِّزات في كل عاصمة أو مدينة كبيرة ، حسب أهميتها التجارية . هكذا ، فالمستعمل في باريس يمكنه العمل على الحاسب . في أوهايو ، وذلك بطلب رقم شبكة مدينة من باريس وبواسطة أي مركز تلفوني ، مُجهَّز بأداة طرفية مع أداة سمعية . المستعمل الآخر الموجود في الطرف الآخر للولايات المتحدة يمكنه أن يطلب نفس الحاسب الموجود في Cleveland ، وذلك بطلب المُركِّز الموجود مثلاً في لوس أنجلوس ، بنفس الطريقة . وفي النهاية بإمكان كل من



شكل 21 ـ شبكة متعددة من نوع نجمة

المستعملين أن يستعمل نفس السجلات الموجودة على الحاسب : وهذا يعطي كثيراً من الفوائد والإمكانيات في حالة الشركات البعيدة ، وفي مستوى إدارتها .

7.3 ـ الشبكات الزردية . مثال عن الشبكة TRANSPAC

الشبكة TRANSPAC هي شبكة عامة لإرسال المعطيات ، طورتها مؤسسة التلغراف في فرنسا P.T ، وترتكز على تبديل الرزم المؤمنة بواسطة مجموعة حاسبات مُتخصّصة أو المبدلات المرتبطة فيا بينها بواسطة وصلتين على الأقل بسرعة عالية حوالي 72 Kb تشكلان إذاً شبكة زردية . بين اثنين(Equipement terminaux de traitement ، E.T.T.D (de donnée تُقسّم المعلومات إلى رزم وتُوجه من خلال الشبكة . الوسائط

المستعملة تؤمن حماية ضد الأخطاء وإمكانية عمل جيدة . إضافة لذلك ،فالشبكة تلعب دوراً إيجابياً في التبديل .والمعايرة في المستوى الدولي والذي تتوافق معه TRANSPAC سيسمح من جهة أخرى بتوصيلها مع شبكات أخرى بتبديل بالـرزم ، مثلاً الشبكـة الأوروبية EURONET ، TELENENT (أميركا) أو DATAPAC (كندا) .

7.3.1 ـ البروتوكول النموذجي لبلوغ TRANSPAC

البلوغ إلى الشبكة TRANSPAC يتم بتوصيل ETTD مع مُبدِّل . تبادل الرزم على هذه الوصلة يتم في مستوى البروتوكول X 25 للجنة CCITT . من الممكن تمييز ثـلاثة مستويات في إدارة هذه الوصلة .

المستوي الفيزيائي :

تُـوصَّـل ETTD مع المبـدِّل بواسـطة وصلة خاصـة بأربعـة خيوط في إزدواجيـة متكاملة . الوصلات تتوافق مع الإعلان V 24 أو V 25 للجنة CCITT .

مستوى السلّـم

الرزمة المطلوب إرسالها هي القسم معلومات من سلّـم من نوع HDLC (الفقرة 6.3) . إجراء التبادل بـين ETTD والمُبدِّل أو LAP (Ligne Access procedure) هـو إجراء من نوع HDLC متوازن . وبشكل خاص ، هناك إكتشاف للأخطاء المنقولة بواسطة FCS ب 16 بتة وترقيم بواسطة نظام بترقيم مزدوج مستقل . الأجراء الأكثر تعقيداً ويدعى متعدد الخطوط ، يسمح بإدارة التبادل بين ETTD والأدوات على عدة توصيلات ، مما يزيد من الأمان . في محيط المعيار X 25 ، الإمكانية هي ممكنة بالنسبة للأدوات ،غير المتكيّفة مع HDLC لاستعمال ، في هذا المستوى ، الإجراءات القديمة التزامنية في مستـوى السمة (يُقال « في الصيغة الأساسية » أو ECMA 24 مثل BCC ، TMM) الخ .

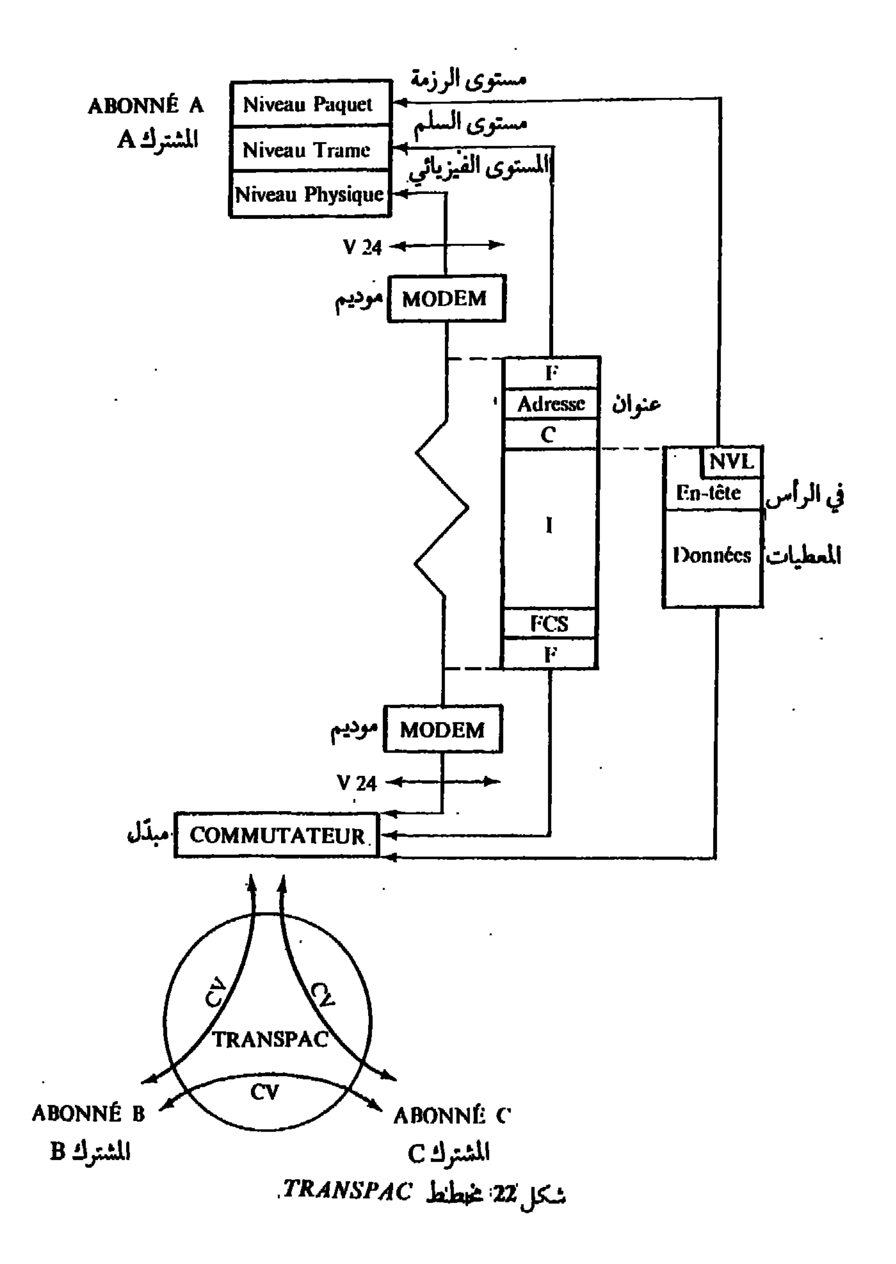
مستوى الرزمة

في هذا المستوى ، المفهوم الأساسي هو مفهوم الدارة الافتراضية أو Circuit CV) (Virtuel ، والتي تعني مسار توجيه مسار الرزم بين أداتين ETTD من خلال الشبكة . الدارة الافتراضية يمكن أن تبدَّل (CVC) ، أي أن يتم التبديل أو يُحرَّر حسب الطلب ، أو دائمة (CVP) ، بطريقة التوصيلة الخاصة .

عدة دارات CV مستقلة بين نفس ETTD وأدوات أخرى مناسبة يُمكن أن تكون مربوطة بمعبر على نفس الوصلة TRANSPAC . يتعلّـق ذلك ببلوغ متعدد الخطوط . سيتم مراجعة مختلف الخطوط بواسطة رقم الخط المنطقي NVL والحاضر على الرزمة . في مستوى الدارات الافتراضية ، والامكانيات المقدّمة هي :

ـ إجراء أو تحرير CVC .

Converted by Tiff Combine - (no stamps are applied by registered version)



شكل 22: مخطط *TRANSPAC*.

ـ إنتقال رزم المعلومات على طول CV مع إحتفاظ بنظام الإرسال . هنا أيضاً يسمح ترقيم مزدوج في مستوى الرزمة بالتدقيق في الدفق المعلوماتي في المحطة المُستقبلة .

ـ الرزمة الخاصة أو « إنقطاع » يُمكن أن ترسل بشكل مستقل عن التدقيق . بالترتيب وبالدفق .

ـ الرزم « إعداد » تسمح بترك المسار الجاري كاملاً على CV .

- الرزمة « معاودة » تؤدي إلى تحرير جميع الدارات CVC وإعداد جميع الدارات CVP للوصلة .

7.3.2 - البلوغ في الصيغة اللاتزامنية

بطبيعتها نفسها ، لا يُمكن للأدوات الطرفية اللاتزامنية أن تستعمل البروتوكول النموذجي X 25 المرتكز على HDLC . مهايئة هذه الأدوات الطرفية بالشبكة يتم في مستوى المبدِّل بواسطة «PAD» أو برنامج - جمع - بعثرة الرزم . إضافة إلى جمع - بعثرة السمات في الرزم ، البرنامج PAD يدير الحوار مع الأداة الطرفية التي تسمح بإقامة ومراقبة الدارة CV . بعض الاعلانات من CCITT (X 29, X 28, X 3) توصف جميع هذه المهام . يمكن للأداة الطرفية اللاتزامنية أن تبلغ إلى المبدِّل بواسطة الشبكة المبدِّلة ، التلكس أو مؤكداً بواسطة وصلة خاصة .

7.3.3 - تعريف الشبكة

تعريف الشبكة TRANSPAC هو مستقل عن الوضع الجغرافي للمُشترك ، ويحتوي أساساً على إشتراك شهري يتعلّق بسرعة البلوغ وتعرفة خاصة حسب حجم المعطيات المرسلة .

مراجع

كتب مرجعية :

C. MACCHI, J. F. GUILBERT, *Téléinformatique; transport et traitement de l'information dans les réseaux et systèmes téléinformatiques*, Dunod-Informatique (réédition 1981).

On pourra aussi consulter :

AFNOR-ISO, Publications relatives à la téléinformatique, à l'AFNOR, Paris.

UK POST OFFICE, *Handbook of Data Communications*, NCC Publication, Manchester, UK, 1975.

D. W. DAVIS, D. L. A. BARBER, *Communication Networks for Computers*, J. Willey & Son (1973).

L. POUZIN, *Réseaux d'ordinateurs/Computer Networks, First European Workshop* (Arles, 1973) (Document publié par l'I.R.I.A.) (1973).

M. ELIE, H. ZIMMERMANN, Une approche systématique des protocoles sur un réseau d'ordinateurs, *Congrès AFCET de* 1973. (Document disponible auprès de l'AFCET, à Paris.)

TRANSPAC, Les documents relatifs à TRANSPAC auprès de la Société TRANSPAC, Tour Maine-Montparnasse, BP 145, 33, avenue du Maine, Paris.

Rapport NORA et ses annexes (annexe 1 en particulier).

ملحقات

A ـ كود ASCII وEBCDIC للمكنات بباية .
B ـ جداول التكويد الداخلي للسمات لمختلف المكنات .
C ـ قوة 2 ، من 0 إلى 64 .
D ـ ألفباء المعلوماتية .

الملحق A1 ـ جدول ASCII - EBCDIC

Dec. عشري	Hex سادس عشري	EBCDIC Code Carte كود تثقيب	Binaire ثنائي	EBCDIC	ASCII	Dec. عشري	Hex سادس عشري	EBCDIC Code Carte كود تثقيب	Binaire ثنائي	EBCDIC	ASCII
0	00	12-0-1-8-9	0000 0000	NUL	NUL	32	20	11-0-1-8-9	0010 0000	DS	SP
1	01	12-1-9	0000 0001	SOH	SOH	33	21	0-1-9	0010 0001	SOS	!
2	02	12-2-9	0000 0010	STX	STX	34	22	0-2-9	0010 0010	FS	"
3	03	12-3-9	0000 0011	ETX	ETX	35	23	0-3-9	0010 0011		#
4	04	12-4-9	0000 0100	PF	EOT	36	24	0-4-9	0010 0100	BYP	$
5	05	12-5-9	0000 0101	HT	ENQ	37	25	0-5-9	0010 0101	LF	%
6	06	12-6-9	0000 0110	LC	ACK	38	26	0-6-9	0010 0110	ETB	&
7	07	12-7-9	0000 0111	DEL	BEL	39	27	0-7-9	0010 0111	ESC	'
8	08	12-8-9	0000 1000		BS	40	28	0-8-9	0010 1000		(
9	09	12-1-8-9	0000 1001		HT	41	29	0-1-8-9	0010 1001		)
10	0A	12-2-8-9	0000 1010	SMM	LF	42	2A	0-2-8-9	0010 1010	SM	*
11	0B	12-3-8-9	0000 1011	VT	VT	43	2B	0-3-8-9	0010 1011	CU2	+
12	0C	12-4-8-9	0000 1100	FF	FF	44	2C	0-4-8-9	0010 1100		,
13	0D	12-5-8-9	0000 1101	CR	CR	45	2D	0-5-8-9	0010 1101	ENQ	-
14	0E	12-6-8-9	0000 1110	SO	SO	46	2E	0-6-8-9	0010 1110	ACK	.
15	0F	12-7-8-9	0000 1111	SI	SI	47	2F	0-7-8-9	0010 1111	BEL	/
16	10	12-11-1-8-9	0001 0000	DLE	DLE	48	30	12-11-0-1-8-9	0011 0000		0
17	11	11-1-9	0001 0001	DC1	DC1	49	31	1-9	0011 0001		1
18	12	11-2-9	0001 0010	DC2	DC2	50	32	2-9	0011 0010	SYN	2
19	13	11-3-9	0001 0011	TM	DC3	51	33	3-9	0011 0011		3
20	14	11-4-9	0001 0100	RES	DC4	52	34	4-9	0011 0100	PN	4
21	15	11-5-9	0001 0101	NL	NAK	53	35	5-9	0011 0101	RS	5
22	16	11-6-9	0001 0110	BS	SYN	54	36	6-9	0011 0110	UC	6
23	17	11-7-9	0001 0111	IL	ETB	55	37	7-9	0011 0111	EOT	7
24	18	11-8-9	0001 1000	CAN	CAN	56	38	8-9	0011 1000		8
25	19	11-1-8-9	0001 1001	EM	EM	57	39	1-8-9	0011 1001		9
26	1A	11-2-8-9	0001 1010	CC	SUB	58	3A	2-8-9	0011 1010		:
27	1B	11-3-8-9	0001 1011	CUI	ESC	59	3B	3-8-9	0011 1011	CU3	;
28	1C	11-4-8-9	0001 1100	IFS	FS	60	3C	4-8-9	0011 1100	DC4	<
29	1D	11-5-8-9	0001 1101	IGS	GS	61	3D	5-8-9	0011 1101	NAK	=
30	1E	11-6-8-9	0001 1110	IRS	RS	62	3E	6-8-9	0011 1110		>
31	1F	11-7-8-9	0001 1111	IUS	US	63	3F	7-8-9	0011 1111	SUB	?

الملحق A2 ـ الجدول EBCDIC - ASCII

Dec.	Hex	EBCDIC Code Carte	Binaire	EBCDIC	ASCII	Dec.	Hex	EBCDIC Code Carte	Binaire	EBCDIC	ASCII
64	40		0100 0000	Sp	@	96	60	11	0110 0000		
65	41	12-0-1-9	0100 0001		A	97	61	0-1	0110 0001		a
66	42	12-0-2-9	0100 0010		B	98	62	11-0-2-9	0110 0010		b
67	43	12-0-3-9	0100 0011		C	99	63	11-0-3-9	0110 0011		c
68	44	12-0-4-9	0100 0100		D	100	64	11-0-4-9	0110 0100		d
69	45	12-0-5-9	0100 0101		E	101	65	11-0-5-9	0110 0101		e
70	46	12-0-6-9	0100 0110		F	102	66	11-0-6-9	0110 0110		f
71	47	12-0-7-9	0100 0111		G	103	67	11-0-7-9	0110 0111		g
72	48	12-0-8-9	0100 1000		H	104	68	11-0-8-9	0110 1000		h
73	49	12-1-8	0100 1001		I	105	69	0-1-8	0110 1001		i
74	4A	12-2-8	0100 1010		J	106	6A	12-11	0110 1010		j
75	4B	12-3-8	0100 1011		K	107	6B	0-3-8	0110 1011		k
76	4C	12-4-8	0100 1100	<	L	108	6C	0-4-8	0110 1100	%	l
77	4D	12-5-8	0100 1101	(	M	109	6D	0-5-8	0110 1101	–	m
78	4E	12-6-8	0100 1110	+	N	110	6E	0-6-8	0110 1110	>	n
79	4F	12-7-8	0100 1111	¦	O	111	6F	0-7-8	0110 1111	?	o
80	50	12	0101 0000	&	P	112	70	12-11-0	0111 0000		p
81	51	12-11-1-9	0101 0001		Q	113	71	12-11-0-1-9	0111 0001		q
82	52	12-11-2-9	0101 0010		R	114	72	12-11-0-2-9	0111 0010		r
83	53	12-11-3-9	0101 0011		S	115	73	12-11-0-3-9	0111 0011		s
84	54	12-11-4-9	0101 0100		T	116	74	12-11-0-4-9	0111 0100		t
85	55	12-11-5-9	0101 0101		U	117	75	12-11-0-5-9	0111 0101		u
86	56	12-11-6-9	0101 0110		V	118	76	12-11-0-6-9	0111 0110		v
87	57	12-11-7-9	0101 0111		W	119	77	12-11-0-7-9	0111 0111		w
88	58	12-11-8-9	0101 1000		X	120	78	12-11-0-8-9	0111 1000		x
89	59	11-1-8	0101 1001		Y	121	79	1-8	0111 1001		y
90	5A	11-2-8	0101 1010	!	Z	122	7A	2-8	0111 1010		z
91	5B	11-3-8	0101 1011	$	[	123	7B	3-8	0111 1011		{
92	5C	11-4-8	0101 1100	*	/	124	7C	4-8	0111 1100		:
93	5D	11-5-8	0101 1101	)	]	125	7D	5-8	0111 1101		}
94	5E	11-6-8	0101 1110	;	¬	126	7E	6-8	0111 1110		–
95	5F	11-7-8	0101 1111	¬	–	127	7F	7-8	0111 1111		DEL

EBCDIC - ASCII ملحق A3 ـ الجدول

Dec.	Hex	EBCDIC Code Carte	Binaire	EBCDIC	ASCII	Dec.	Hex	EBCDIC Code Carte	Binaire	EBCDIC	ASCII
128	80	12-0-1-8	1000 0000			160	A0	11-0-1-8	1010 0000	-	
129	81	12-0-1	1000 0001	a		161	A1	11-0-1	1010 0011	a	
130	82	12-0-2	1000 0010	b		162	A2	11-0-2	1010 0010	s	
131	83	12-0-3	1000 0011	c		163	A3	11-0-3	1010 0011	t	
132	84	12-0-4	1000 0100	d		164	A4	11-0-4	1010 0100	u	
133	85	12-0-5	1000 0101	e		165	A5	11-0-5	1010 0101	v	
134	86	12-0-6	1000 0110	f		166	A6	11-0-6	1010 0110	w	
135	87	12-0-7	1000 0111	g		167	A7	11-0-7	1010 0111	x	
136	88	12-0-8	1000 1000	h		168	A8	11-0-8	1010 1000	y	
137	89	12-0-9	1000 1001	i		169	A9	11-0-9	1010 1001	z	
138	8A	12-0-2-8	1000 1010			170	AA	11-0-2-8	1010 1010		
139	8B	12-0-3-8	1000 1011	(		171	AB	11-0-3-8	1010 1011	└	
140	8C	12-0-4-8	1000 1100	≤		172	AC	11-0-4-8	1010 1100	┘	
141	8D	12-0-5-8	1000 1101	(		173	AD	11-0-5-x	1010 1101	[	
142	8E	12-0-6-8	1000 1110	→		174	AE	11-0-6-8	1010 1110	≥	
143	8F	12-0-7-8	1000 1111	+		175	AF	11-0-7-8	1010 1111	•	
144	90	12-11-1-8	1001 0000			176	B0	12-11-0-1-8	1011 0000	0	
145	91	12-11-1	1001 0001	j		177	B1	12-11-0-1	1011 0001	1	
146	92	12-11-2	1001 0010	k		178	B2	12-11-0-2	1011 0010	2	
147	93	12-11-3	1001 0011	l		179	B3	12-11-0-3	1011 0011	3	
148	94	12-11-4	1001 0100	m		180	B4	12-11-0-4	1011 0100	4	
149	95	12-11-5	1001 0101	n		181	B5	12-11-0-5	1011 0101	5	
150	96	12-11-6	1001 0110	o		182	B6	12-11-0-6	1011 0110	6	
151	97	12-11-7	1001 0111	p		183	B7	12-11-0-7	1011 0111	7	
152	98	12-11-8	1001 1000	q		184	B8	12-11-0-8	1011 1000	8	
153	99	12-11-9	1001 1001	r		185	B9	12-11-0-9	1011 1001	9	
154	9A	12-11-2-8	1001 1010			186	BA	12-11-0-2-8	1011 1010		
155	9B	12-11-3-8	1001 1011	)		187	BB	12-11-0-3-8	1011 1011	┘	
156	9C	12-11-4-8	1001 1100			188	BC	12-11-0-4-8	1011 1100	┐	
157	9D	12-11-5-8	1001 1101	)		189	BD	12-11-0-5-8	1011 1101	]	
158	9E	12-11-6-8	1001 1110	±		190	BE	12-11-0-6-8	1011 1110	+	
159	9F	12-11-7-8	1001 1111	=		191	BF	12-11-0-7-8	1011 1111	–	

EBCDIC - ASCII ملحق A4 ـ الجدول

Dec.	Hex	EBCDIC Code Carte	Binaire	EBCDIC	ASCII	Dec.	Hex	EBCDIC Code Carte	Binaire	EBCDIC	ASCII
192	C0	12-0	1100 0000			224	E0	0-2-5	1110 0000		
193	C1	12-1	1100 0001	A		225	E1	11-0-1-9	1110 0001		
194	C2	12-2	1100 0010	B		226	E2	0-2	1110 0010	S	
195	C3	12-3	1100 0011	C		227	E3	0-3	1110 0011	T	
196	C4	12-4	1100 0100	D		228	E4	0-4	1110 0100	U	
197	C5	12-5	1100 0101	E		229	E5	0-5	1110 0101	V	
198	C6	12-6	1100 0110	F		230	E6	0-6	1110 0110	W	
199	C7	12-7	1100 0111	G		231	E7	0-7	1110 0111	X	
200	C8	12-8	1100 1000	H		232	E8	0-8	1110 1000	Y	
201	C9	12-9	1100 1001	I		233	E9	0-9	1100 1001	Z	
202	CA	12-0-2-8-9	1100 1010			234	EA	11-0-2-8-9	1110 1010		
203	CB	12-0-3-8-9	1100 1011			235	EB	11-0-3-8-9	1110 1011		
204	CC	12-0-4-8-9	1100 1100			236	EC	11-0-4-8-9	1110 1100		
205	CD	12-0-5-8-9	1100 1101			237	ED	11-0-5-8-9	1110 1101		
206	CE	12-0-6-8-9	1100 1110			238	EE	11-0-6-8-9	1110 1110		
207	CF	12-0-7-8-9	1100 1111			239	EF	11-0-7-8-9	1110 1111		
208	D0	11-0	1101 0000			240	F0	0	1111 0000	0	
209	D1	11-1	1101 0001	J		241	F1	1	1111 0001	1	
210	D2	11-2	1101 0010	K		242	F2	2	1111 0010	2	
211	D3	11-3	1101 0011	L		243	F3	3	1111 0011	3	
212	D4	11-4	1101 0100	M		244	F4	4	1111 0100	4	
213	D5	11-5	1101 0101	N		245	F5	5	1111 0101	5	
214	D6	11-6	1101 0110	O		246	F6	6	1111 0110	6	
215	D7	11-7	1101 0111	P		247	F7	7	1111 0111	7	
216	D8	11-8	1101 1000	Q		248	F8	8	1111 1000	8	
217	D9	11-9	1101 1001	R		249	F9	9	1111 1001	9	
218	DA	12-11-2-8-9	1101 1010			250	FA	12-11-0-2-8-9	1111 1010		
219	DB	12-11-3-8-9	1101 1011			251	FB	12-11-0-3-8-9	1111 1011		
220	DC	12-11-4-8-9	1101 1100			252	FC	12-11-0-4-8-9	1111 1100		
221	DD	12-11-5-8-9	1101 1101			253	FD	12-11-0-5-8-9	1111 1101		
222	DE	12-11-6-8-9	1101 1110			254	FE	12-11-0-6-8-9	1111 1110		
223	DF	12-11-7-8-9	1101 1111			255	FF	12-11-0-7-8-9	1111 1111		

ملحق B1 ـ جدول التكويد لشركة IBM

سمة	كود داخلي ستة عشري	كود التثقيب	السمة	كود داخلي ستة ـ عشري	كود التثقيب
A	C1	12-1	6	F6	6
B	C2	12-2	7	F7	7
C	C3	12-3	8	F8	8
D	C4	12-4	9	F9	9
E	C5	12-5	+	4E	12-8-6
F	C6	12-6	–	60	11
G	C7	12-7	*	5C	11-8-4
H	C8	12-8	/	61	0-1
I	C9	12-9	(	4D	12-8-5
J	D1	11-1	)	5D	11-8-5
K	D2	11-2	$	5B	11-8-3
L	D3	11-3	=	7E	8-6
M	D4	11-4	blanc ou espace	40	sans perforation
N	D5	11-5			
O	D6	11-6		6B	0-8-3
P	D7	11-7		4B	12-8-3
Q	D8	11-8			
R	D9	11-9	(quote)	7D	8-5
S	E2	0-2	<	4C	12-8-4
T	E3	0-3	>	6E	0-8-6
U	E4	0-4	≠	7B	8-3
V	E5	0-5	—	6D	0-8-5
W	E6	0-6			
X	E7	0-7	|	4F	12-8-7
Y	E8	0-8	&	50	12
Z	E9	0-9	;	5E	11-8-6
0	F0	0	:	7A	8-2
1	F1	1	¬	5F	
2	F2	2	?	6F	0-8-7
3	F3	3			
4	F4	4	!	5A	11-8-2
5	F5	5	%	6C	0-8-4

ملحق B2 ـ جدول التكويد CDC للسلسلة CYBER 70

سمة	كود داخلي ثماني	كود تثقيب	سمة	كود داخلي ثماني	كود تثقيب
A	01	12-1	0	33	0
B	02	12-2	1	34	1
C	03	12-3	2	35	2
D	04	12-4	3	36	3
E	05	12-5	4	37	4
F	06	12-6	5	40	5
G	07	12-7	6	41	6
H	10	12-8	7	42	7
I	11	12-9	8	43	8
J	12	11-1	9	44	9
K	13	11-2	+	45	12
L	14	11-3	–	46	11
M	15	11-4	*	47	11-4-8
N	16	11-5	/	50	0-1
O	17	11-6	(	51	0-4-8
P	20	11-7	)	52	12-4-8
Q	21	11-8	$	53	11-3-8
R	22	11-9	=	54	3-8
S	23	0-2	espace ou blanc	55	sans perforation
T	24	0-3		56	0-3-8
U	25	0-4		57	12-3-8
V	26	0-5	]	62	0-2-8
W	27	0-6	(quote)	64	4-8
X	30	0-7	<	72	12-0
Y	32	0-8	>	73	11-7-8
Z	32	0-9	;	77	12-7-8

ملحق B3 ـ جدول التكويد UNIVAC للسلسلة 1110 / 1108

سمة	كود داخلي ثماني	كود تثقيب	سمة	كود داخلي ثماني	كود تثقيب
A	06	12-1	0	60	0
B	07	12-2	1	61	1
C	10	12-3	2	62	2
D	11	12-4	3	63	3
E	12	12-5	4	64	4
F	13	12-6	5	65	5
G	14	12-7	6	66	6
H	15	12-8	7	67	7
I	16	12-9	8	70	8
J	17	11-1	9	71	9
K	20	11-2	)	40	12-4-8
L	21	11-3	—	41	11
M	22	11-4	+	42	12
N	23	11-5	<	43	12-6-8
O	24	11-6	=	44	3-8
P	25	11-7	>	45	6-8
Q	26	11-8	$	47	11-3-8
R	27	11-9	*	50	11-4-8
S	30	0-2	(	51	0-4-8
T	31	0-3		56	0-3-8
U	32	0-4	'(quote)	72	4-8
V	33	0-5	;	73	11-6-8
W	34	0-6	/	74	0-1
X	35	0-7	.	75	12-3-8
Y	36	0-8	blanc ou espace	05	sans perforation
Z	37	0-9			

ملحق B4 ـ جدول التكويد BCD
(مثلاً CII HONYWELL-BULL)

سمة	كود داخلي ثماني	كود تثقيب	سمة	كود داخلي ثماني	كود تثقيب
A	21	12-1	0	00	0
B	22	12-2	1	01	1
C	23	12-3	2	02	2
D	24	12-4	3	03	3
E	25	12-5	4	04	4
F	26	12-6	5	05	5
G	27	12-7	6	06	6
H	30	12-8	7	07	7
I	31	12-9	8	10	8
J	41	11-1	9	11	9
K	42	11-2	)	55	11-5-8
L	43	11-3	–	52	11
M	44	11-4	+	60	12-0
N	45	11-5	<	36	12-6-8
O	46	11-6	=	75	0-5-8
P	47	11-7	>	16	6-8
Q	50	11-8	$	53	11-3-8
R	51	11-9	:	54	11-4-8
S	62	0-2	(	35	12-5-8
T	63	0-3	,	73	0-3-8
U	64	0-4	(quote)	57	11-7-8
V	65	0-5	;	56	11-6-8
W	66	0-6	/	61	0-1
X	67	0-7	.	33	12-3-8
Y	70	0-8	blanc ou espace	20	sans perforation
Z	71	0-9			
			!	77	0-7-8
				32	12

ملحق C ـ جدول قوة 2

Positives			
1	0	4,294,967,296	32
2	1	8,589,934,592	33
4	2	17,179,869,184	34
8	3	34,359,738,368	35
16	4	68,719,476,736	36
32	5	137,438,953,472	37
64	6	274,877,906,944	38
128	7	549,755,818,888	39
256	8	1,099,511,627,776	40
512	9	2,199,023,255,552	41
1,024	10	4,398,046,511,104	42
2,048	11	8,796,093,022,208	43
4,096	12	17,592,186,044,416	44
8,192	13	35,184,372,088,832	45
16,384	14	70,368,744,177,664	46
32,768	15	140,737,488,355,328	47
65,536	16	281,474,976,710,656	48
131,072	17	562,949,953,421,312	49
262,144	18	1,125,899,906,842,624	50
524,288	19	2,251,799,813,685,248	51
1,048,576	20	4,503,599,627,370,496	52
2,097,152	21	9,007,199,254,740,992	53
4,194,304	22	18,014,398,509,481,984	54
8,388,608	23	36,028,797,018,963,968	55
16,777,216	24	72,057,594,037,927,936	56
33,554,432	25	144,115,188,075,855,872	57
67,108,864	26	288,230,376,151,711,744	58
134,217,728	27	576,460,752,303,423,488	59
268,435,456	28	1,152,921,504,606,846,976	60
536,870,912	29	2,305,843,009,213,693,952	61
1,073,741,824	30	4,611,686,018,427,387,904	62
2,147,483,648	31	9,223,372,036,854,775,808	63

اللائحة رقم 1

بلوغ مباشر Acces direct — direct access, random access

طريقة في الكتابة أو في قراءة المعطيات تتم بواسطة عناوين تدل على مكانتها .

بلوغ متوالي Acces séquentiel — Serial access

طريقة لقراءة أو كتابة المعطيات ، تتم حسب ترتيب محدَّد سابقاً .

مقطوع ، مستقل Autonome — off-line

يقال عن عتاد يعمل باستقلالية عن عتاد آخر .

بنك معطيات Banque de données — data bank

مجموعة من المعطيات المناسبة لحقل معين يُعرَّف عن إستعلامات مرتبة بطريقة تسمح باستشارتها بواسطة المستعملين .

مجمع معطيات Base de données — data base

مجموعة معطيات منظَّمة لجهة إستعمالها بواسطة برامج مناسبة لتطبيقات محدَّدة ويطريقة يسهل معها التطوير المستقل للمعطيات والبرامج .

معلوماتية مكتبية Bureautique

مجموعة التقنيات والوسائط التي تؤدي إلى أتمتة الأعمال المكتبية ويشكل أساسي لمعالجة التبادل بالمعلومات بواسطة الكلام ، الكتابة ، والصور .

توافقية Compatibilité

نوعية عتاد ومناهج متوافقة مع قواعد ملقى نظام معلوماتي مُحدَّد ، وحيث الإدخال لا يُغيِّر شروط تشغيل النظام .

إسطوانة ممغنطة Disque magnetique

أسطوانة مغطاة بطبقة مغناطيسية تسجَّل عليها المعلومات .

أسطوانة ضوئية Disque optique

أسطوانة لتسجيل معلومات مقروءة بوسائط ضوئية .

إسطوانة مرنة Disquette — Floppy disk

إسطوانة صغيرة مرنة ، بسعة خزن مختزلة .

معطيات — Donnée — Data

تمثيل للمعلومات بشكل توافقي يساعد على معالجتها .

على الخط — En ligne — on line

يُقال عن عتاد عندما يعمل بعلاقة مباشرة مع عتاد آخر .

معلوماتية — Genie informatique

مفاهيم ، وإجراءات تتعلّق بنظام معلوماتي

زيادة — Increment

الكمية التي تزيد من قيمة متحولة في كل طور من تنفيذ البرنامج .

معالجة الرسوم — Infographique

تطبيق المعلوماتية في معالجة الرسوم والصور .

معلومات — Information

عنصر معرفة قادر على أن يكون مُمثلاً بواسطة أرقام أو إشارات كي يتم تخزينه ، معالجته أو إرساله .

معلوماتية — Informatique

علوم معالجة المعلومات بواسطة مكنات أوتوماتيكية . كناقل الاستعلامات البشرية ، والتبادل في الحقول التقنية ، الاقتصادية والاجتماعية .

تعليمة — Instruction — Statement, instruction

جملة في لغة برمجة تقوم بعملية معينة .

تخاطبي — Interactif

لتمييز عتاد ، برامج أو شروط إستغلال نظام معلوماتي يسمح بالردود المعاكسة والإجابة في صيغة تخاطبية مع المستعملين وفي الوقت الفعلي .

حلقي — Interface

وصلة بين أداتين أو مناهج تسمح بتبادل المعلومات مع إعتبار القواعد المشتركة ، الفيزيائية أو المنطقية .

إصدار لائحة — Listage

أ ـ وثيقة مطبوعة بواسطة طابعة الحاسب .

ب ـ فعل طبع أصدر لائحة .

طبع ، أصدر لائحة ، عرض — Lister

أ ـ إصدار وثيقة متواصلة بواسطة طابعة الحاسب .

ب ـ تقديم معطيات أو تعليمات .

مناهج — Logiciel — Software

مجموعة البرامج ،العمليات والقواعد ، ومحتملاً الوثائق ، المناسبة لتشغيل مجموعة معالجة المعطيات .

عتاد — Materiel — Handware

مجموعة العناصر الفيزيائية المستعملة في معالجة المعطيات .

ذاكرة — Memoire

عنصر يسمح بتسجيل وحفظ وإستخراج المعلومات

ذاكرة خزن بحجم كبير — Memoire de masse

الذاكرة الخارجية ذات السعة الكبيرة في الخزن .

ذاكرة ميتة ، ذاكرة ثابتة — Memoire masse — Read only memory

ذاكرة لا يمكن تعديل مضمونها في الاستعمال العادي (ROM) .

ذاكرة حيّـة ، ذاكرة قراءة ـ كتابة — Memoire vive,

ذاكرة يمكن تعديل مضمونها في الاستعمال العادي (RAM)

معالج صغري ، ميكروبروسسور — Microprocesseur

معالج صغير ، حيث جميع عناصره مجموعة في دارة تكاملية واحدة .

صيغة تخاطبية — Mode dialogué

طريقة في معالجة المعطيات تسمح بالحوار بين المستعمل والنظام .

برمجة متعدِّدة — Multiprogrammation

تقنية في العمل تسمح بتنفيذ عدة برامج في وقت واحد .

معالجة متعددة — Multitraitement

طريقة في تشغيل الحاسب ، بحيث تستطيع عدة معالجات بلوغ ذاكرة مشتركة تعمل بالتوازي على برامج مختلفة .

رقمي — Numerique — numeric

على عكس نظيري ، وهو يعني تمثيل المعطيات أو الكميات الفيزيائية بواسطة سمات ، عبارة عن أرقام . النظام الذي يعالج المعلومات الرقمية هو نظام معالجة رقمية .

قسمة الوقت — Partage de temps — time sharing

تقنية إستغلال نفس الحاسب بواسطة عدة مستعملين تقوم بتنفيذ أعمال مستقلة في صيغة عمل تخاطبية .

قلم ضوئي — Photo style — light pen

أداة إدخال ، يُؤشـر بواسطتها المُؤشر مباشرة على الشاشة .

منقول . Partabilité

يُقال عن برنامج صالح للاستعمال على عدة أنظمة معلوماتية من أنواع مختلفة .

معالج Processeur

أ ـ العضو من الحاسب الذي يقوم بتفسير وتنفيذ التعليمات .

ب ـ مجموعة البرامج التي تسمح بتنفيذ برامج مكتوبة في لغة معينة على الحاسب .

مُعالجات متعددة Bi-, Tri-, multiprocesseur

حاسب يحتوي على أكثر من معالج مركزي .

مناهج تطبيقية progiciel

مجموعة كاملة ومُوثقة من البرامج المعتمدة للاستعمال بواسطة عدة مستعملين، لجهة نفس العمل التطبيق ونفس المهمة .

روبوت Robotique

مجموعة للدراسات والتقنيات التي تؤدي إلى تقليد حواس البشر .

للنجاة De secours

تعني الاجراءات والعتاد الخاص بالاستعمال في حالة حدوث عطل طارىء معين عند التشغيل .

مشرف ، قائد ـ Serveur

مجموعة تقوم باستغلال نظام معلوماتي تسمح للطالب باستشارة وإستعمال مباشر لواحد أو عدة بنوك للمعطيات .

نظام التشغيل Système d'exploitation

المناهج والبرامج التي تدير الحاسب ، وتكون مستقلة عن البرامج التطبيقية ، ولكنها ضرورية لوضع هذه الأخيرة في العمل وإدارة عمليات الادخال والإخراج .

معلوماتية بعيدة Teleinformatique

إستغلال أوتوماتيكي لنظام معلوماتي يستعمل شبكات الاتصال البعيدة .

معلوماتية بعيدة Telematique

مجموعة الخدمات المعلوماتية المقدّمة بواسطة شبكة إتصالات بعيدة .

معالجة بعيدة Teletraitement

طريقة في المعالجة تتم بواسطتها إرسال وإستقبال المعلومات بواسطة أدوات طرفية بعيدة عن الحاسب .

معالجة بعيدة بالحصص Teletraitement par lots

المعالجة البعيدة التي تحتوي على تجميع بالحصص للبرامج المطلوب تنفيذها أو المعطيات المطلوب معالجتها .

وقت فعلي — Temps réel — reel time

طريقة معالجة تسمح بقبول المعطيات في لحظة معينة والحصول المباشر على النتائج .

أداة طرفية — Terminal

جهاز يسمح ببلوغ نظام معلوماتي من بعيد .

نسخة مطبوعة — Tirage — fas-sin

وثيقة رسمية ناتجة عن الإرسال على ناقل دائم لصورة محضرة للعرض .

معالجة أوتوماتيكية للمعطيات — Traitement automatique des données

مجموعة العمليات الجارية بوسائط أوتوماتيكية ، مناسبة لتجميع ، تسجيل ، تعديل ، حفظ ، تدمير ، تنقيح المعطيات وبشكل عام إستغلالها .

معالجة بالحصص — Traitement par lots

طريقة لتشغيل المعطيات تتم بواسطتها تجميع البرامج والمعطيات المطلوب معالجتها في حصص .

عرض ، عرضي — Visu, Visuel — display

أداة تسمح بتمثيل مرئي للمعلومات .

عرض — Visualiser — to display

كتابة النتائج المعالجة على أداة عرض (شاشة) .



فهرس

الموضوع — الصفحة







هذا الكتاب

يعالج هذا الكتاب المواضيع العملية اليومية التي يصطدم بها المعلوماتي : ولقد جرى إختيار العناصر والمواضيع الأكثر إستعمالاً والأكثر فائدة ، مع الإبتعاد عن المسائل النظرية أو الشديدة التخصّص .

ولقد رأينا من الأنسب أن يُعالج هذا الكتاب أولاً موضوع تمثيل وتنظيم المعلومات والمعطيات ، وبعد ذلك جرى إيجاز وتقديم المفاهيم الأساسية بالنسبة للسجلات ، وبشكل عام تلك التي تخص بنوك المعطيات .

القسم العملي في المعلوماتية هو المُتعلّق بالبرمجة . لذلك يُوجد قسم أساسي يتعلّق بالأربعة لغات الأكثر إستعمالاً وهي : BASIC ، PL / 1, COBOL, FORTRAN 4, 77 . بينما وضعنا لغة المؤول (assembler) جانباً لأنها تتعلّق بنوعية المكنة المستعملة ، نفس الشيء بالنسبة لأنظمة التشغيل : هنا يلزم مساعد عملي خاص بالنظام حيث يقوم كل مصمّم بإصدار نسخة خاصة به وحده . وبما إن تطور المعلوماتية البعيدة أصبح كبيراً ، لذلك خصّصنا فصلاً لها يعالج التقنيات المستعملة فيها .

يعتبر هذا الكتاب مفيداً لجميع الطلاب الراغبين في دراسة المعلوماتية ، كما هو مفيد للاختصاصيين العاملين في هذا الحقل حيث يساعدهم على بلوغ ما يريدونه من المعلومات دون الحاجة إلى البحث في الكتب والمراجع ، لذا فهو مساعداً عملياً في المعلوماتية .

وفي النهاية يحتوي هذا الكتاب على قاموس بالمصطلحات العلمية المستعملة في حقل الكومبيوتر .

